

Zimbabwe Electricity Transmission and Distribution Company
Fault Reporting and Tracking Android Application



TATENDA CHIKOSI (R134508Z)

Zimbabwe Electricity Transmission and Distribution Company
Fault Reporting and Tracking Android Application



By
Tatenda Chikosi (R134508Z)

Submitted in partial fulfilment of the requirements for the degree of

BSc Honours Information Systems

Department of Computer Science and Information Systems

in the

Faculty of Science and Technology

at the

Midlands State University

Gweru

November 2017

Supervisor: **Mr. M Giyane**

ABSTRACT

The core aspect behind developing Zimbabwe Electricity Transmission and Distribution Company Fault reporting and tracking android application was a result of inefficiency of the manual system that was used at the organization. The main aim was to develop a computerized system that will curb the problems currently encountered by using the existing manual system by integrating information systems process. The instruments which were used to come up with the system include MySQL database, notepad ++, android studio and xampp. Various feasibility studies were carried out to identify if it is worthy to build the new system. Feasibility studies has shown that the available hardware and software can be used with the new system without facing any challenges and also that the benefits of the new system outweigh its costs. Information gathering methodologies were used to identify how the old system works. These include interviews, questionnaires and observations. The system was designed by making much emphasis on the architectural, physical, database and logical design. The system was tested using various testing techniques which encompasses black box and white box testing. Validation and verification processes were carried out and proved that the system meets that expected standards. The system was finally installed using the direct conversion method. The system needs to be maintained for it to provide best results. The users of the system are recommended to stick to the basics learnt during the training, and should they face any problems, they should consult the system administrator.

DECLARATION

I CHIKOSI TATENDA (R134508Z) do hereby declare that I am the sole author of this dissertation. I authorize the Midlands State University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

Signature.....

Date

APPROVAL

This dissertation “Zimbabwe Electricity Transmission and Distribution Company Fault Reporting and Tracking Android Application” by **Chikosi Tatenda** meets the regulation governing the award of the degree of **BSC Information Systems Honours Degree** by Midlands State University and is approved for its contribution to knowledge and literary presentation.

Supervisor.....

Date.....

ACKNOWLEDGEMENTS

I foremost give praise to the Lord Jesus, for it is by his grace for me to reach this far. Special thanks dedicated to my supervisor Mr M. Giyane who initiated the direction and support through the course period of my project research. I likewise want to thank all my lecturers for imparting knowledge of tremendous value to me. I would also want to thank Mr and Mrs Chikosi for their financial support, may God bless you. Many thanks to my friends and relatives for the encouragement and help they offered amid the entire course of my project. I would like also to thank the management and employees at ZETDC for their unwavering support and information that they offered to make this project a success.

DEDICATION

This research project is dedicated to my parents for their unwavering support through their financial assistance. I would like to express my sincere gratitude to all my friends and relatives who have been a pillar of strength during good and bad times and their continued support throughout the course of the project.

TABLE OF CONTENTS

PAGE

Contents

ABSTRACT.....	1
DECLARATION.....	ii
APPROVAL.....	iii
ACKNOWLEDGEMENTS.....	iv
DEDICATION.....	v
TABLE OF CONTENTS.....	vi
LIST OF ACRONYMS.....	xi
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
LIST OF APPENDICES.....	xiv
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Background of study.....	1
1.2.1 Background of Organisation.....	1
1.2.2 Organisational structure.....	2
1.2.3 Vision.....	4
1.2.4 Mission statement.....	4
1.3 Problem definition.....	4
1.4 Aim.....	4
1.5 Objectives.....	4
1.6 Instruments and methods.....	5
1.6.1 Instruments.....	5
1.6.2 Data collection methods.....	5
1.7 Justification and rationale.....	6
1.8 Conclusion.....	6
CHAPTER 2: PLANNING PHASE.....	7

2.1 Introduction.....	7
2.2.1 Reasons for developing the system	8
2.3 Feasibility study	8
2.3.1 Technical feasibility.....	8
2.3.1.1 Hardware and software requirements	9
2.3.1.2 Technical expertise	10
2.3.2 Economic feasibility	10
2.3.2.1 Tangible benefits.....	10
2.3.2.2 Intangible benefits	11
2.3.2.3 Development costs	12
2.3.2.4 Operational costs	12
2.3.2.5 Cost Benefit Analysis.....	13
2.3.2.6 Return on Investment (ROI).....	14
2.3.2.7 Net Present Value (NPV)	14
2.3.2.8 Payback period.....	15
2.3.3 Social feasibility.....	15
2.3.3.1 Management.....	15
2.3.3.2 Employees.....	15
2.3.3.3 Clients.....	16
2.3.3.4 Overview of the social feasibility	16
2.3.4 Operational feasibility.....	16
2.3.4.1 Operational feasibility overview	16
2.3.5 Risk analyses	16
2.3.5.1 Technical risk.....	17
2.3.5.2 Stakeholders analysis.....	17
2.3.5.3 Risk planning	18
2.4 Work plan.....	18
2.4.1 Gantt chart	19
2.5 Conclusion	19
CHAPTER 3: ANALYSIS PHASE.....	20
3.1 Introduction.....	20
3.2 Information Gathering Methodologies	20
3:2:1 Observation.....	20

3:2:2 Interview	21
3:2:3 Questionnaire	22
3.3 Analysis of existing system	23
3.3.1 Description of the current system	23
3.3.1.1 System Input	23
3.3.1.2 Processes.....	23
3.3.1.3 Output.....	23
3.4 Process analysis	23
3.4.1 Activity diagram	24
3.5 Data analysis	26
3.5.1 Context diagram.....	26
3.5.2 Data flow diagram.....	27
3.6 Weaknesses of the current system	29
3:7 Evaluation of alternatives	29
3:7:1 Outsourcing.....	29
3.7.2 Improvement	30
3.7.3 In house development	30
3.8 Requirement analyses	32
3.8.1 Functional requirements.....	32
3.8.1.1 Front end for clients.....	32
3.8.1.2 Back end for the organization.....	33
3.8.1.3 Use Case diagram.....	33
3.8.2 Non- functional requirements	34
3.9 Conclusion.....	36
CHAPTER 4: DESIGN PHASE.....	37
4.1 Introduction.....	37
4.2 System Design	37
4.2.1 How Fault Reporting and Tracking Android application works	37
4.2.2 Context diagram of the proposed system	38
4.2.3 Data flow diagram of the proposed system.....	39
4.3 Architectural design	40
4.4 Physical design.....	41

4.5 Database Design.....	42
4.5.1.1 External level.....	42
4.5.1.2 Conceptual level.....	42
4.5.1.3 Internal level	43
4:5:2 Logical design.....	43
4.5.3 Enhanced Entity Relationship diagram	45
4.6 Program design	46
4.6.1 Package diagram	47
4.6.2 Class diagram	47
4.6.3 Sequence diagram.....	48
Fig 4.9: Sequence diagram.....	49
Key.....	49
4.7 Pseudo code	50
4.8 Interface design.....	51
4.8.1 Menu design	51
4.8.1.1 Main menu.....	51
4.8.1.2 Sub menus.....	52
4.8.2 Input design	53
4.8.3 Output design	56
4.9 Security design.....	58
4.9.1 Physical security.....	58
4.9.2 Network security.....	59
4.9.3 Operational security.....	59
4.10 Conclusion	59
CHAPTER 5: IMPLEMENTATION PHASE.....	60
5.1 Introduction.....	60
5.2 Coding.....	60
5.3 Testing.....	60
5.3.1 Unit testing.....	61
5.3.1.1 Black box testing	61
5.3.1.2 White box testing.....	61
5.3.2 Module testing.....	62

5.3.3 Subsystem testing.....	62
5.3.4 System testing.....	62
5.3.5 Acceptance testing.....	62
5.3.5.1 Alpha testing	62
5.3.5.2 Beta testing	63
5.3.6 Testing strategies	63
5.3.6.1 Validation	63
5.3.6.2 Verification.....	67
5.4 Installation.....	70
5.4.1 User training.....	71
5.4.2 Data migration	71
5.4.3 Changeover strategies	71
5.4.3.1 Direct changeover	71
5.4.3.2 Parallel changeover.....	72
5.4.3.3 Phased changeover	72
5.4.3.4 Recommended changeover strategy	72
5.5 Maintenance.....	72
5.5.1 Corrective maintenance.....	72
5.5.2 Adaptive maintenance	73
5.5.3 Perfective maintenance	73
5.5.4 Recommendations for maintenance	73
5.6 Recommendations for future development.....	73
5.6.1 Recommendations to stakeholders	74
5.7 Conclusion	74
REFERENCE LIST	75
APPENDICES	78

LIST OF ACRONYMS

ORGANOGRAMOrganizational structure
NPV..... Net Present Value
ROIReturn on investment
SQLStructured query language
ZETDC.....Zimbabwe Electricity Transmission and Distribution Company

LIST OF TABLES	PAGE
Table 2.1 hardware requirement	9
Table 2.2: Software Requirements.....	10
Table 2.3: Tangible benefits	11
Table 2.4: Development costs.....	12
Table 2.5: Operational costs	12
Table 2.6: Cost Benefit Analysis	13
Table 2.7: Net Present Value	14
Table 2.8: payback period.....	15
Table 2.9: Time Schedule	18
Table 2.10: Gantt chart	19
Table 3.1 Alternative cost evaluation	31
Table 4.1: Login Table.....	43
Table 4.2 Fault reporting table.....	44
Table 4.3 Pending faults table.....	44
Table 4.4 Resolved faults table.....	45

LIST OF FIGURES	PAGE
Fig 1.1 Organogram of ZETDC	3
Fig 3.3 Data flow diagram	28
Fig 3.4 Case diagram	34
Fig 4.1 Context diagram	38
Fig 4.3 Architectural design.....	40
Fig 4.4 Physical design	41
Fig 4.5: Database Design	42
Fig 4.6 Enhanced Entity Relationship diagrams.....	46
Fig 4.7 Package diagram.....	47
Fig 4.8 Class diagram	48
Fig 4.9 Sequence diagram.....	Error! Bookmark not defined.
Fig 4.10 Main menu.....	52
Fig 4.11 Sub Menu.....	Error! Bookmark not defined.
Fig 4.12 Log in.....	54
Fig 4.13 Creating user account	54
Fig 4.14 Report fault	55
Fig 4.15 Assign task.....	55
Fig 4.16 Feedback.....	56
Fig 4.17 Faults	57
Fig 4.18 Notification.....	58
Fig 4.19 Fault status feedback	58
Fig 5.1 Testing Process	61
Fig 5.2 User creation validation.....	64
Fig 5.3 Login validation.....	64
Fig 5.4 Creating user account	65
Fig 5.5 Tracing of fault	65
Fig 5.6 User access level.....	66
Fig 5.7 Data integrity	67
Fig 5.8 System locking	67
Fig 5.9 System objective 1.....	68
Fig 5.10 System objective 2.....	69
Fig 5.11 System objective 3.....	69
Fig 5.12 System objective 4.....	70
Fig 5.13 System objective 5.....	70

LIST OF APPENDICES

PAGE

Appendix A: User manual.....	80
Appendix B: Interview checklist.....	90
Appendix C: Questionnaire score sheet.....	92
Appendix D: Observation	93
Appendix E: Snippet of code	94

CHAPTER 1: INTRODUCTION

1.1 Introduction

ZESA is the only company that provides electricity in Zimbabwe. The organization's background, vision and the mission statement will be discussed in this phase. This chapter mainly focuses on fault reporting and tracking system of the company. The company is having a challenge in managing and tracking electricity faults reported by its customers. This chapter will also discuss problems faced in the current system and objectives of the proposed system i.e. solutions to overcome the problems faced by the company.

1.2 Background of study

Dubois, Jolibort and Muhlbacher (2007) defined the background of study as the research phase that provides essential description of the research to be discussed on a particular topic. The current systems have many problems to the organisation such as there is high rate of customer complaints as most faults are left unattended due failure of cooperation of the depot clerk and to forgetting to tell those on duty to go and attend the fault. The faults are recorded in a counter book. However, a new system is being proposed to overcome the problems faced within the current system. The main objective of the project is to design a fault reporting and tracking android application that enables the depot clerk to log client faults in the system and also allow clients to report faults and comments directly on the application. The application is going to manage client queries, tracking faults as they are resolved and publishing notices on that application. The android application will regularly update the fault status of a particular client.

1.2.1 Background of Organisation

According to ZESA Holdings (1985), the company was formed following the passing of two Acts by the Parliament of Zimbabwe was the Electricity Act (Chapter 13:20) comprehensive reforms have since been experienced in the electricity house. The Electricity Act also ushered in the formation of five successor companies, the Zimbabwe Power Company, Zimbabwe Electricity Transmission Company, Zimbabwe Electricity Distribution Company, ZESA Enterprises and Powertel Communications. In line with the approved structure, all power generation assets and operations are under ZESA Holdings generation subsidiary, the Zimbabwe Power Company was formed in October 1996.

Distribution assets and supply functions fall under the Zimbabwe Electricity Distribution Company, ZESA Enterprise, and another subsidiary of ZESA Holdings comprises of four

business units namely ZESA Technology Centre, Production and Services, Transport Logistics and Projects. It is a flexible investment arm for ZESA Holdings that has a diversified business portfolio. Transmission grid assets and operations fall under the Zimbabwe Electricity Transmission Company. A separate Rural Electrification Fund, a Board and Agency have been set up under the Rural Electrification Fund Act, passed in January 2002. The Agency is administering a special fund formed under this Act to finance rural electrification projects. The Board is accountable to the Minister of Energy and Power Development.

1.2.2 Organisational structure

Gray and Larson (2003), defines an organisational structure as a hierarchical chart showing clearly the flow of information, coordination and delegation of tasks in achieving organizational goals. It provides the motive for evaluating employee performance. There are many organisational structure such as divisional, matrix and functional structures. Martin (1998) define divisional organisational structure as way to group employees that have the same product type. Matrix organisational structure is when many employees report to more than one manager, which means there will be multiple reporting lines within the organisation (Martin, 1998). Functional organisational structure divides the organisation in many subgroups that specialises in a specific area/ department. ZETDC uses the functional organizational structure as illustrated in fig 1.1.

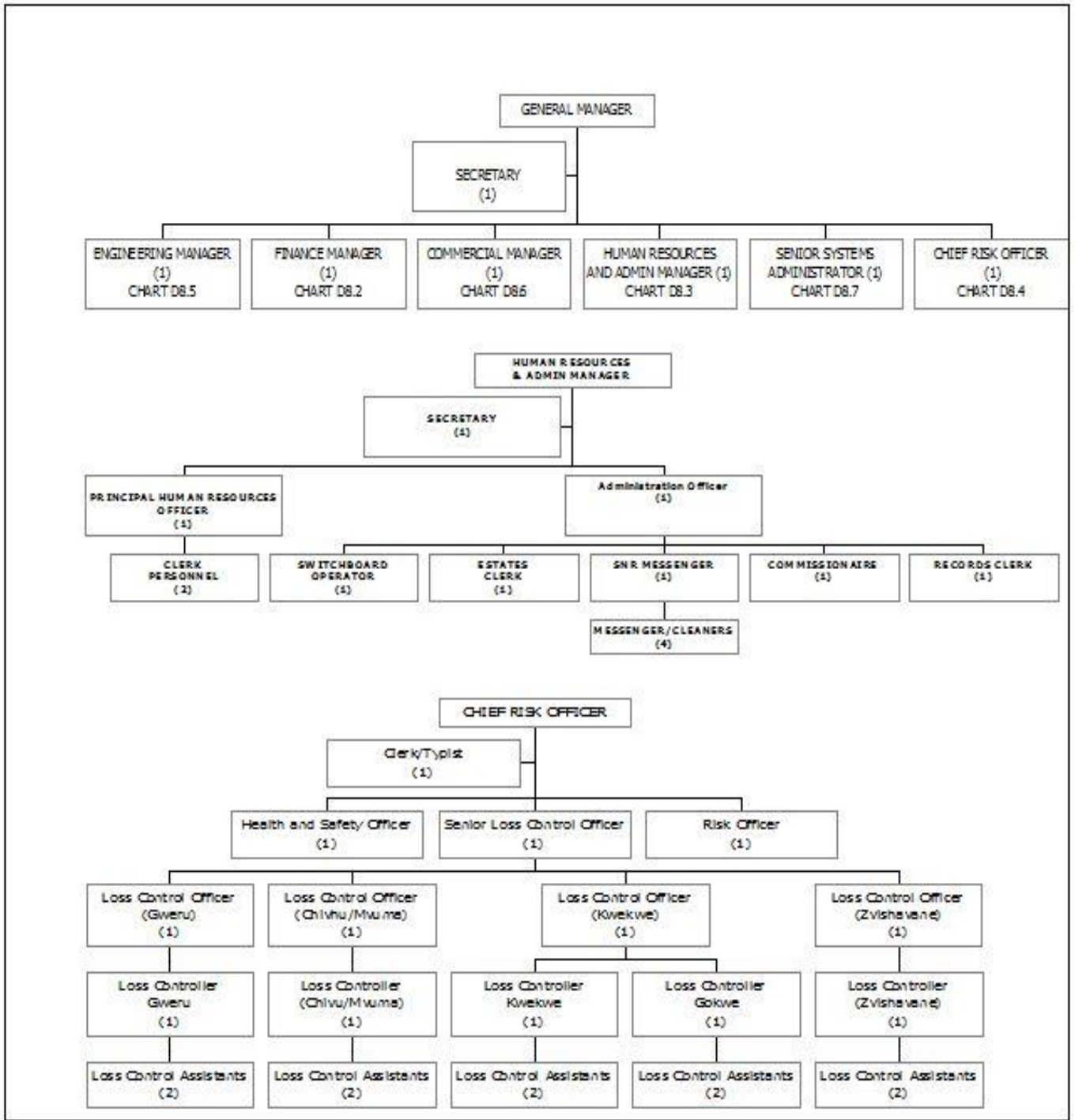


Fig 1.1 Organogram of ZETDC

1.2.3 Vision

Dubois et al. (2007) define a vision statement as a core purpose in which organisation seeks to accomplish in its operational life time. It acts as a guideline of what action to take in the future course of the organisation. The vision of Zimbabwe Electricity Transmission and Distribution Company is to be the preferred provider of electricity regionally and related services globally.

1.2.4 Mission statement

Needham and Dransfield (1997) state that a mission statement is a declaration written to support organisation's focus and its core purpose that remains unchanged for a long time. The mission statement of Zimbabwe Electricity Transmission and Distribution Company is to bring convenience to our valued customers through the provision of adequate, safe, reliable electricity and related services at competitive prices

1.3 Problem definition

Currently, faults are recorded in books which is not secure to the organisation. Some faults are left unattended because the depot clerk will forget to log on faults in the book. It also takes time for a fault to be responded to and there is no a forum for customers to give their suggestions. As a result of there will be high customer complaints and this will tarnish the image of the organisation. And also there is no time management in attending faults as artisans will forge hours when attending the fault. The Android Fault Reporting and Tracking application will solve these problems and also provide a forum for client queries and enhance feedback to both the artisan and the clients.

1.4 Aim

Jewell (2005) defines an aim as set goals that organisations want to achieve. It is also defined as the overall purpose of the study. The main of this study is to develop a Fault Reporting and Tracking Android application.

1.5 Objectives

According to Zelman (2009), objective provides a clear explanation of the project's outcome. Clearly defined objectives guarantee organisation success and it gives a sense of purpose to every within the organisation. The objectives for ZETDC Fault Reporting and Tracking Android application are as follows:

- To enable instant response as soon as electricity faults are reported on the application.
- To allow the depot foreman to allocate faults to artisans.

- To allow the customer to track the fault so as to trace the fault status.
- To enable the artisan to update fault status after attending the fault.
- To alert the depot foreman on pending faults through a pop up notification.
- To enhance customer relationship management through an open forum where people can comment or give suggestions and also to inform the general public on how to conserve electricity.

1.6 Instruments and methods

The proposed system is developed to overcome the problems faced within the current system and the following instruments and data collections methods are used to develop the proposed system.

1.6.1 Instruments

- **MySQL** – Cornell (2008) state that MySQL it is an open source database management software used for database creation. It is efficient in data manipulation such as searching or deleting of data in a database. This software is easy to learn, hold larger databases and is compatible with various operating systems.
- **PHP** - Meloni (2004) states that it is a java based open source platform that allows a software developer to create a customised integrated development environment (IDE). It supports plug-ins that enables the programmer to develop and test code written in other languages.
- **Android Studio** - it is the flexible gradle-based system IDE used to develop applications for android phones and tablets based on the Intelligent IDE.
- **Notepad ++** - it is the best manuscript editor that facilitates a good text file creation and to open them.
- **Java** -java is an object oriented language to be used in development of the view of the application on the ZETDC administrative side of the application.

1.6.2 Data collection methods

The following three methods highlighted will be used to gather information on how the current manual system is operated.

- **Interviews**

McMullan (2009) defined an interview as a shared conversation which is carried out to obtain specific information in line with the research objectives. Interviews carried out were mostly directed to the management team.

➤ **Observation**

McMullan (2009) asserts that observation entails observing how the system is performing by either overt or covert. Observation was carried out to see how the current system actually operates.

➤ **Questionnaire**

McMullan (2009) defined questionnaires as structured questions written down by the examiner and directed to the respondents so as to get essential information to solve a particular problem. Questionnaires were directed to both staff and management.

1.7 Justification and rationale

The proposed android application will a connection between the customer and the organisation enables a strong bond and hence increases customer satisfaction as their queries will be addressed in short period of time. Users of the application will have ease of access as long as there is an internet able device thus physical visits to the depot to report critical faults or request for their faults to be attended will be unnecessary. Through the forum provided by the application, clients can track on whether the faults are attended or not. The proposed system is designed to minimise the workload as well as the use of paper work since everything will be automated.

1.8 Conclusion

This chapter has clearly managed to state all the problems of the existing system such as failing to manage faults reported by customers. It also highlights the solutions to the problems identified above and also the development tools of the proposed system. The next chapter will enlighten on whether this project is feasible or not.

CHAPTER 2: PLANNING PHASE

2.1 Introduction

This chapter is important in project research because it shows or gives direction to the developers of work to be done. It also follows up the progress of project including its activities. Feasibility study is done in this phase to evaluate or examine the project viability considering the technical, social, economic and operational feasibility. This phase will also highlight the business value of the proposed system. Analysis of risks involved will also be addressed and a work plan showing various activities that are going to take place in developing the project.

2.2 Business Values

O'Brien (2005) defines business value as the concepts that expands the value and benefits of a firm beyond economic value thus customer, supplier and managerial value. These business values help a firm to achieve its goals since it benefits both the clients and the organisation as a whole. The business values of the proposed ZETDC Fault reporting and tracking system are as follows:

a) Customers

- Customer satisfaction – better relations will be improved between the customers and the company since customers will be now directly report their faults on the application anytime and hence this increase will greatly satisfy the customers and this will maintain high standards of quality service.
- Increased efficiency – The proposed will be efficient in tracking of faults since it will be easy as clients are able to report faults anytime and are responded to quickly and resolved immediately.

b) Management

- Cost reduction – All client's details will be now stored in the organisation's database as many clients as possible. This improves data security and results in less use of paper of recording faults.
- Goodwill of the business in terms of efficiency and effectiveness.
- System will also benefit the management as they will automatically get monthly fault reports any time upon requesting to the system.

2.2.1 Reasons for developing the system

The proposed system will help ZETDC to manage faults that are reported by its valuable customers. The system will enable customers to report their faults anytime and also allows them to provide feedback to the organisation pertaining the service provided to them. The system will provide information to the lead artisan thus feedback from the customers and this feedback will be used to analyse customer complaints and able to make proper decisions on what action to take so as to satisfy our valuable clients. The reasons for developing the proposed system are as follows:

- To enable the customers to report their faults anytime anywhere as long there is internet access.
- The system will help the managers and lead artisan to manage and track the faults attended to and not.
- The system will generate monthly reports that are used by managers to analyse faults reported and able to see unattended faults.
- The proposed system provides an effective two-way communication between the organisation and its valuable customers.
- The system will easily help customers to access the services about the organisation's notices thus warning of electricity usage, shutdown schedules and a forum for customers to provide complaints and complements.

2.3 Feasibility study

Hoffer, George and Valacich (2002) defines feasibility study as an assessment or examination of a project considering its technical, economic, social and operational feasibility. It also evaluates the project's potential for success.

2.3.1 Technical feasibility

This involves checking if the organisation has the technical requirements to develop a project up to its completion (Kendall and Kendall, 2008). Therefore, a technical feasibility should be considered before embarking on developing a system. It considers the software, hardware and the necessary staff required to use the system and also to analyse if the available technical retirements are able to suit with the proposed system.

2.3.1.1 Hardware and software requirements

Hardware requirements

Table 2.1 shows the hardware requirements that are needed for the development of the ZETDC Fault reporting and tracking android application.

Table 2.1 hardware requirement

Hardware required	Available	Quantity required	Decision to be done
HP Laser printer	No	1	ZETDC is willing to buy the printer
Router	Yes		
Backup server	No	1	ZETDC is willing to buy the server
Switch	Yes		
HP laptop	No	1	ZETDC is willing to buy the requested printers
Wireless modem	No	1	ZETDC is willing to buy the modem
Tablet mobile phone	No	2	ZETDC is willing to purchase the tablet phone
Network cables	60 metres	100 metres	ZETDC is willing to buy the requested network cables

Software requirements

Table 2.2 shows the software requirements that are needed for the development of the ZETDC Fault reporting and tracking android application.

Table 2.2: Software Requirements

Software required	Available	Quantity required	Decision to be done
Eclipse	No	1	Open source software
Android studio	No	1	Open source software
Java (NetBeans)	No	1	Open source software
MySQL Server	Yes		

2.3.1.2 Technical expertise

The ZETDC project is technically feasible to carry out because the organisation is willing and able purchase hardware and software requirements to develop the system. The system will be easy to implement because the organisation already has its IT expertise so there is no need for training the users.

2.3.2 Economic feasibility

The way to determine positive benefits derived from the proposed system to the organisation economically (McBride, 2002). It also highlights developmental and operational costs of the proposed system and also expected benefits will be evaluated. Economic feasibility also focuses on the assessment of the cost benefit analyses of the proposed system thus if benefits outweigh costs then it's feasible to carry on with the project.

2.3.2.1 Tangible benefits

Benefits that are quantifiable and real and these benefits are easily assessable, measurable and justified (Hoffer et al. 2002). Table 2.3 shows the benefits for using the proposed system:

Table 2.3: Tangible benefits

Benefit	Monetary value
Reduced labour	1750
Reduced paperwork	500
Better worker throughput	500
Improvement in management planning	250
Total	3000

2.3.2.2 Intangible benefits

This is the accumulation of benefits after a specified period of time and are not easily quantified or measurable (O'Brien, 2005). The intangible benefits that are derived from using the system are as follows:

- Improved the delivery of services to all system users.
- There will be better customer satisfaction
- Increased goodwill within the organisation
- Efficiency in work performance
- Quality and fast in report generation and decision making

2.3.2.3 Development costs

Expenditures experienced during the development of the project.

Table 2.4: Development costs

Item	Quantity	Amount
Wireless modem	1	50
Backup server	1	675
Tablet mobile phone	2	210
Laser jet printer	1	45
Hp Laptop	2	225
Network cable	100 m	15
Total		1220

2.3.2.4 Operational costs

Costs that are experienced on daily basis whenever an organisation is operation (Kendall and Kendall, 2011). Table 2.5 shows the costs that will be incurred when the system is now in operation:

Table 2.5: Operational costs

Operational costs	Value/USD Year 1	Value/USD Year 2	Value/USD Year 3
Software upgrades	50	160	180
Internet charges	80	140	530
Software maintenance	100	140	100
Computer consumables	50	100	120
Training costs	70	60	20
Total costs	350	600	950

2.3.2.5 Cost Benefit Analysis

The process of assessing the system costs and benefits to capital invested to it in particular at a particular point in time (Riley, 2012). It is essential in all projects because it enables the organisation to tell whether if it is feasible to carry on considering the costs and a budgeted amount for a particular project. It defines the feasibility of the project in financial terms and benefits. Return on investment, net present value and payback period are the tools that are going to be used show the economic benefit of the proposed system. Table 2.6 shows the cost benefit analyses:

Table 2.6: Cost Benefit Analysis

Costs	Year 1		Year 2		Year 3	
Operational costs	\$ 350		\$ 600		\$ 950	
Development costs	1220		-		-	
		1520		600		950
Total cost		(1520)		(600)		(950)
Benefits						
Tangible	3000		4000		5000	
Intangible						
Total benefits	-		-		-	
Net profit		3000		4000		5000
		1480		3400		4050

The cost benefit analysis above indicates that the project is viable to carry on because it has noted that the benefits outweigh the costs.

2.3.2.6 Return on Investment (ROI)

Randall (1996) states that it is an appraisal technique that provides an easy narrative to non-accounting user of information and helps in selecting the project with high ROI. Below are the calculations of ROI:

The formula is:

$$\begin{aligned}\text{Return on Investment} &= \frac{\text{Total benefits} - \text{total costs}}{\text{Total benefits}} * 100 \\ &= \frac{35\,000 - 16\,400}{35\,000} * 100 \\ &= \mathbf{53.1\%}\end{aligned}$$

The ROI is 53.1% and this indicates a favorable investment thus shows the project is feasible economically.

2.3.2.7 Net Present Value (NPV)

The evaluation technique that consider the time value of money thus a dollar today is not dollar tomorrow (Laudon and Laudon, 2013). It shows the cash flow and an investment that yields a positive NPV will be considered or chosen. It is shown in the table 2.7:

Table 2.7: Net Present Value

Year	Discounting factor	Cash flow	Present value
0	1	(7400)	(7400)
1	0.909	1000	909
2	0.826	2000	1652
3	0.751	3000	2253
NPV			2586

The positive value shown in Table 2.7 shows the project's feasibility as it is favorable to the organisation.

2.3.2.8 Payback period

Randall (1996) explains payback period as the time taken to break even to payback the initial investment. The project with the shortest payback period will be chosen on the basis that the organisation wish to minimize the time the project will be in debt.

The table below show the payback period.

Year	Cash flow (\$)	Balance
0	(7400)	(7400)
1	1000	6400
2	3000	3400
3	3400	

Table 2.8: payback period

The organisation will realize the payback after a period of 3 years thus favorable to the organisation.

2.3.3 Social feasibility

Gibson and Hughes (1994) define it as the process of evaluating the benefits that the proposed system provides to the organisation as well as the society as a whole. Its main focus is on examining the impact of the new system to both the organisation and the society. It emphasises on how the proposed system affects the management, employees and customers. There will be an increase in computer literacy rate in the country as customers will able to learn to use technology so to report their faults.

2.3.3.1 Management

The proposed system will positively affect the management because they are now able to track faults reported by customers anytime. This system will make life easier for them to make quick decision since the system will be producing monthly reports which are then used to analyses customers trend and to see the faults that are left unattended.

2.3.3.2 Employees

Employees will greatly benefit from the proposed system. These benefits include less work will be done and there is also reduction in paper use and as a result of this, employee's morale will increase. There will going to be training programs to the end users of the system

so as to mimic resistance from the users and an individual can communicate with clients without even coming to the work.

2.3.3.3 Clients

The development of proposed system will benefit customers a lot since it will be ensuring efficiency and effectiveness in service delivery. It also reduces customer expenses such as coming to the organisation in person and thus, the operation of the proposed system will positively affect the customers. Customers will now be able to express their views on the service rendered by the organisation.

2.3.3.4 Overview of the social feasibility

The above analyses show that the system has a positive impact to the organisation and all of its users. However, the project was considered socially feasible as it yields more positive effects as compared to negatives.

2.3.4 Operational feasibility

Gibson and Hughes (1994) define operational feasibility as the degree at which how problems are solved by the proposed system and able to satisfy the system requirements considering its operating environment. It focuses on the adaptability of the system to suit the organisational operating environment and able to meet the goals stated as well as the ability of system users to support the proposed system. The system will provide the following:

- Backup facility which allows the organisation information to be stored in a separate safe place to prevent data loss.
- There will be great satisfaction of employees i.e. users of the system through training programs which helps to be familiar with the new system.
- The generation of reports by the system helps managers in quality decision making.
- It provides a good graphic user interface which will be user friendly and enables data security.

2.3.4.1 Operational feasibility overview

The system shows positive benefits derived from using the system by various stakeholders.

2.3.5 Risk analyses

Risk analysis is the process of analysing the extent in which the effects of risks will hinder the operation of the new project, (Whitten, Bentley and Dittman, 2004). Risk analysis is a

vital aspect in all projects as it will involve strategic planning which will encounter to prevent the hazards that are likely to affect the running of the system. There are two categories of risk which emphasizing areas where hazards are mostly likely to occur that is technical risk and stakeholders risk. Changes in technology, change of computer programs when they are affected by viruses and resistance to changes are other risks that affects projects.

2.3.5.1 Technical risk

Technical risks such as network problems and viruses will be experienced as the form of risk that will likely to affect the system whilst in operation, (Whitten et al. 2004). The network might have problems in terms of speed, during the initial stages of system use as it will be in use by many users for testing it. The system will face a challenge as it will be striving to suit the working environment with live data.

There are other problems that affect the operation of the system through the use of mobile phones that is if a phone does not have an antivirus program installed in it then it will be greatly affected by viruses which will then cause the malfunctioning of the system.

➤ Mitigation and Management

There should be a close monitoring on network usage by the IT staffs to analyses the network trend and able to increase the bandwidth if there are many users that are using the system so as to improve its performance. Furthermore, the organisation will inform its valuable customers to use effective antivirus protection on their mobile phones and it also protect its servers and computers from threats such as virus or hacking.

2.3.5.2 Stakeholders analysis

The process of analysing the important of certain project to various stakeholders as well as their need (Dennis, Wixom and Roberta, 2012). It influenced with the operations of the organisation either positively or negatively. The research carried out showed that the following are the stakeholders of the proposed system:

- **Users** – These are the ones who play around with the system all the time and interact with it and these include employees and customers. User of the system will play with the system by having access to information, input data, sending comments and feedback to the organisation using the application system. And through this the system may be poor in transmission of information.

- **Management**– These are the ones who plan, allocate resource and control the progress of the system development process. Therefore, if the management does not support the project this will result in project failure in a stipulated specified time frame and may result in poor quality software to be delivered.
- **Employees** – These are the ones who are likely to be inspired by the actions that the organisation will offer. Employees tend to resist change when they think the system will operate differently to what they are used to.

2.3.5.3 Risk planning

This is the formulating of measures that intend to avoid the occurrence of risks. It involves alerting the stakeholders about the risks and informing them about these measures below to mimic the rate of risks:

- **Contingents plan** – these are the strategies which focus on possible future events that one is unable to predict with certainty. Since the risk are unavoidable, they can be prevented. An example will be that of conflicts be team members, since they are unpredictable, they can be dealt with.
- **Minimization strategies** – refers to the measures that help reduce risks, for instance firewalls being place to restrict unauthorized or viruses. The core of risk planning is to diminish the probability of risks through employment of proper strategies.

2.4 Work plan

Gibson and Hughes (1994) define a work plan as the timeframe in which the project activities must take following a sequence and time must be followed as stated. The time allocation of each chapter is shown in table 2.9:

Table 2.9: Time Schedule

Phase	Start	Finish	Duration in weeks
Introduction	19/08/2016	25/08/2016	1
Planning phase	1/10/2016	14/10/2016	2
18 Page			

Analysis phase	25/02/2017	09/03/2017	2
Design phase	20/03/2017	10/04/2017	3
Implementation phase	12/04/2017	19/04/2017	1
Maintenance phase	19/04/2017	Ongoing	

2.4.1 Gantt chart

Gantt chart is a tool that helps with the planning and sequencing of activities that are connected to the objectives of the project, performed out of logical sequence or done twice (Marakas and O'Brien, 2013). This allows the project manager determine the time frame of the activities. The shaded part on the Gantt chart is the time needed at each phase.

Table 2.10: Gantt chart

Week	1	2	3	4	5	6	7	8	9	10	11	12
Project proposal												
Planning												
Analysis												
Design												
Implementation												
Maintenance												
Documentation												

2.5 Conclusion

This phase has managed to evaluate all the requirements of the proposed system and it has found to be feasible to carry on with it. It also managed to address the business values that are derived from using the proposed system. A work plan of how each activity in the development process may take to be completed is defined as well as the analyses of risks that are likely to affect the system. Analyses phase is the next chapter which include the data collection methods and also analysing the current system.

CHAPTER 3: ANALYSIS PHASE

3.1 Introduction

This phase shows a clearly describe how the current system works and identifying requirements that must be considered in developing the proposed system. Processes and operations that are performed within the current system are also discussed in this phase, showing how they link among each other and also analysing the activities that took place in each operation of the current system. Data is collected using various methodologies highlighting its merits and demerits as well as system alternative solutions are evaluated to find the best among a variety of solutions.

3.2 Information Gathering Methodologies

These are techniques are used to capture data required for the development of the system (McMullan, 2009). The research was carried out to gather information about the views and comments of the users about the current system. The data was gathered from various groups of users thus the stakeholders and the employees. The data was found to be important as it helps managers to know stakeholder's requirements. Various methodologies were used collect data such as questionnaires, interviews and observations.

3:2:1 Observation

It refers to the act of observing an individual or a group people unknowingly whilst working (Bluelens, Sinding and Windstorm, 2011). It was observed that the depot clerk was recording faults in the book and also the way customers were handled. Their process involved the use of paper and pen as this slows down the operations. However, customers were to wait for long hours for them to report their faults.

a) Advantages

- Information was obtained without disturbing those who were at work and biases were eliminated.
- There was no intervention with anyone and first-hand information was obtained.

b) Disadvantages

- It was difficult to observe all activities that took place in the process.
- Biased information concerning the behaviour of the staff were obtained as many employees changed their normal working style.

Key findings

Relevant information was observed and noted down as clients spent a lot of time in queues for them to report faults so this was a key aspect. This was because there was use of pen and paper to record faults. Hence an improved system is needed so that customers can report quickly rather than travelling to the depot physically.

3:2:2 Interview

McMullan (2009) defined an interview as a shared conversation which is carried out to obtain specific information in line with the research objectives. This is an effective way of gathering data from various interviewees directly thus face to face or over the phone (Lucas, 2004). The interviewer obtains data through asking a set of questions to the interviewees. Interviewees were providing quick feedback and accurate information which enables the interviewer to draw meaningful conclusions. Interviewees were given a room to ask additional questions which was used for decision making. Various stakeholders were interviewed so as to gather detailed information on how the current system is operating and the opinion of developing a better system. During the interview the following advantages and disadvantages were noted.

a) Advantages

- There was high response rate from the interviews as interviewees attempted to answer all questions that were asked. However, detailed and precise information were obtained as stakeholders used body language to express their views.
- Interviews were efficient as there was further clarification to incomplete answers and there was immediate response from users.

b) Disadvantages

- Some of the stakeholders were not interviewed because they were not available in their respective work places and this wasted a lot of time.
- There was fear in most of the workers as they were not disclosing meaningful information.
- Some interviewees were reluctant to give information.

Key findings

Detailed and accurate information was gathered as interviews help the interviewer to know the problems faced within the current system. This process enabled the interviewer to grasp all stakeholder requirements. A number of customers were complaining about the travelling

cost they incur whenever they are to report a fault. They also indicated that whenever a fault is reported it takes a lot of time to be resolved. Hence, there must be a new system to cater these problems faced within the current system.

3:2:3 Questionnaire

These are a set of questions that are used by researcher to collect information from a sample of population that will be used for making decisions (McMullan, 2009). It is a fact-finding tool that is regarded by many system stakeholders as they will respond to questions without fear of anything. A variety of questionnaires were given to different staff members of different departments and to some of the potential customers. A positive result was noted as respondents took their time to fill in the questions as they were given a period of two weeks.

a) Advantages

- Quality responses were noted as the respondents answer the questions anonymously without fear.
- Questionnaires promoted the element of innovation as respondents were bringing in new facts that helps the researcher to draw a good conclusion.
- Vast information was obtained in a short period of time hence saved time.

b) Disadvantages

- Questionnaires did not provide a room for explanation and this led to misinterpretation of information.
- Some of the questions were not answered well as some respondents failed to interpret the questions.

Key findings

Questionnaires were an effective way because accurate information was obtained as stakeholders were providing their opinions and views without fear. It was indicated that some of the faults were not attended because the depot clerk might forget to log on the fault in the book.

This is so because they might be busy hence the new proposed system will eliminate this error. Stakeholders also indicated the lack of a way for them to give their opinion and comments.

3.3 Analysis of existing system

This stage explains more about how the current fault reporting system works at ZETDC. The current system operates manually as customers travel from various areas to come and report their faults. Thus, when customers want to report faults they have to visit the company physically or by making a phone call. This is a major problem because the depot clerk may forget to log the fault in the book or record wrong information due to pressure of customers. As a result, there will be high rate of customer complains if faults are not attended to. The fault will pass many levels thus if a customer calls or comes physically the fault will be written in the counter book. Then the faults reported will be submitted to the depot foreman for approval and faults will be assigned to artisans to attend the fault. The problem comes if the depot foreman is not around as no faults can be attended without the approval of the depot foreman. The current system is less secure and prone to natural disaster such as fire as faults are logged in the counter book. As a result, data recorded will be lost. However tracking of faults reported will difficult as there is no any backup. Feedback about attended faults is just orally communicated but no record is kept for future reference.

3.3.1 Description of the current system

3.3.1.1 System Input

- ❖ Faults reported
- ❖ Client's details
- ❖ Administrator details

3.3.1.2 Processes

- ❖ File the fault reported
- ❖ Record client details
- ❖ Classify the faults
- ❖ Record in the fault reported list

3.3.1.3 Output

- ❖ Report to the top managers
- ❖ Report to the client
- ❖ list of resolved faults

3.4 Process analysis

Duffy (2011) defines process analysis as the order in which procedures or steps follow each other in forming system activities from start to end of the project. It also shows the processes

involved as well as activities that are done and the movement of data among them. It is used to represent the inputs, outputs of data and processes involved in the current system.

3.4.1 Activity diagram

An activity diagram visually shows a series of actions or flow of control from one activity to another within the system (Felici, 2009). The ZETDC fault reporting and tracking system activity diagram is shown in fig 3.1.

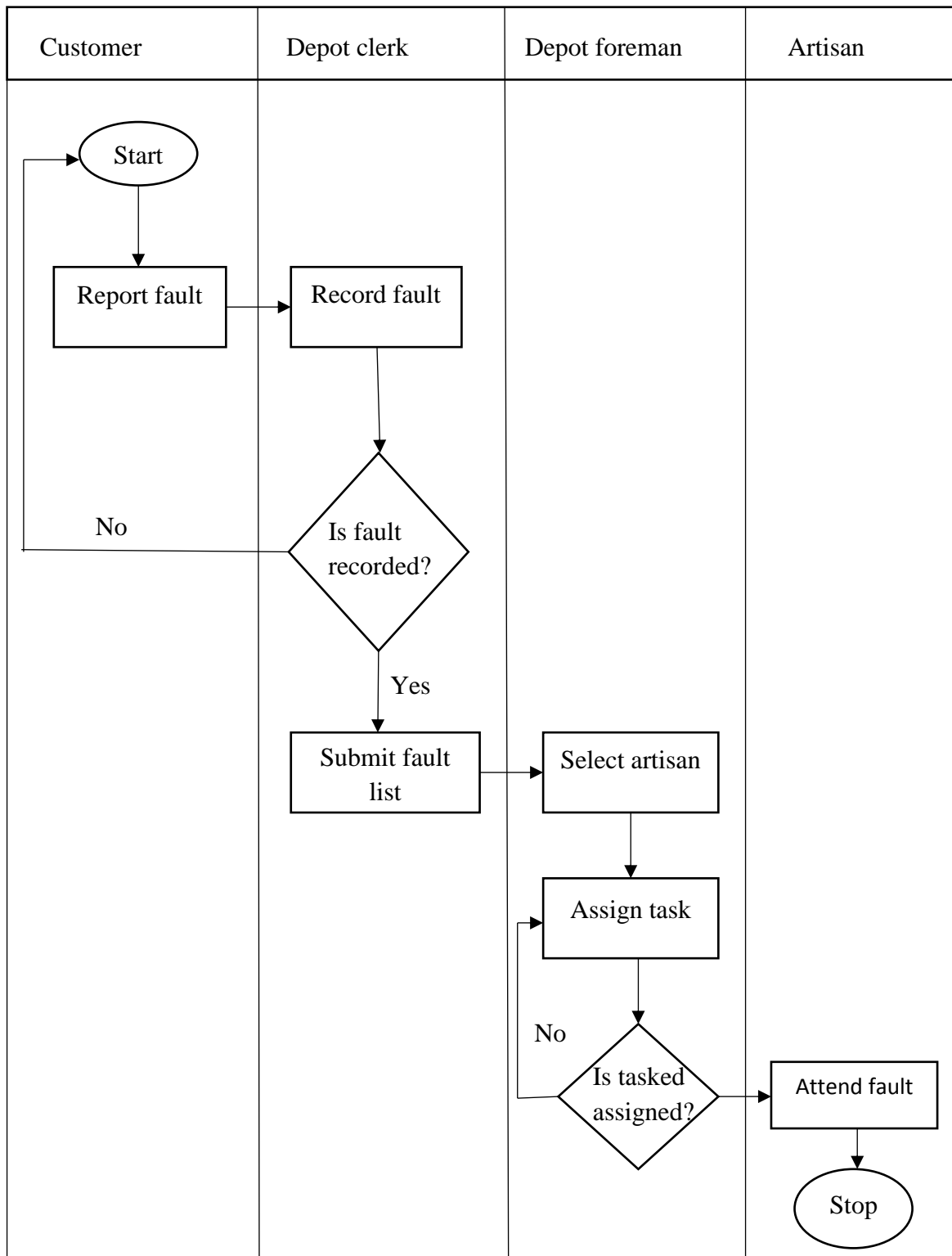
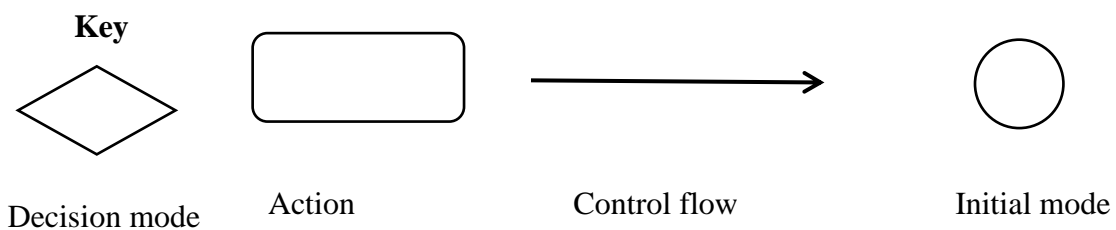


Fig 3.1 Activity diagram



3.5 Data analysis

Kendra (2008) defined data analysis as the process of analysing how data is transformed into useful information that can be used to assist in making decisions. It mainly focuses on managing the process involved in the existing system. This is achieved using context and dataflow diagram.

3.5.1 Context diagram

The diagram show the entities and their relationship with the system that exist in the overall process (Shelly and Rosenblatt, 2010). It consists of many entities and single process and the context diagram of the current ZETDC fault reporting and tracking reflects interactions between clients, depot clerk, depot foreman and artisan and this is shown on fig 3.2.

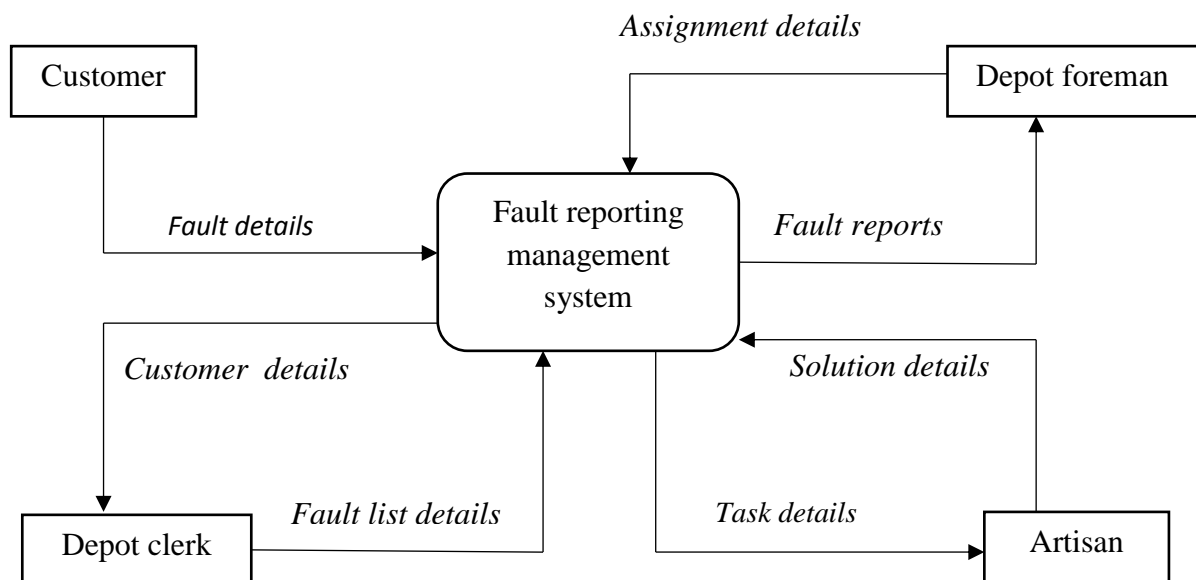
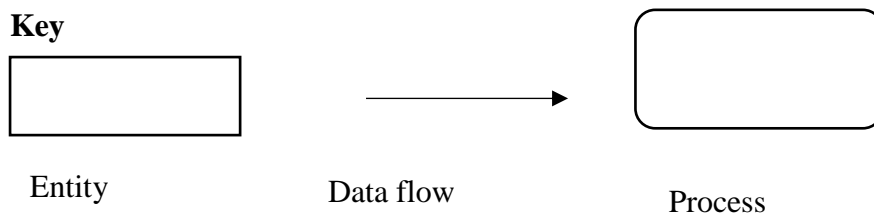


Fig 3.2 Context diagram



3.5.2 Data flow diagram

It is a diagram that designed to show the flow of information from one entity to another and the processes that are involved within the system. It uses a set of tools which are data stores, entity, processes and data flows to clearly represent the operations performed within the system. This is illustrated by the diagram in fig 3.3.

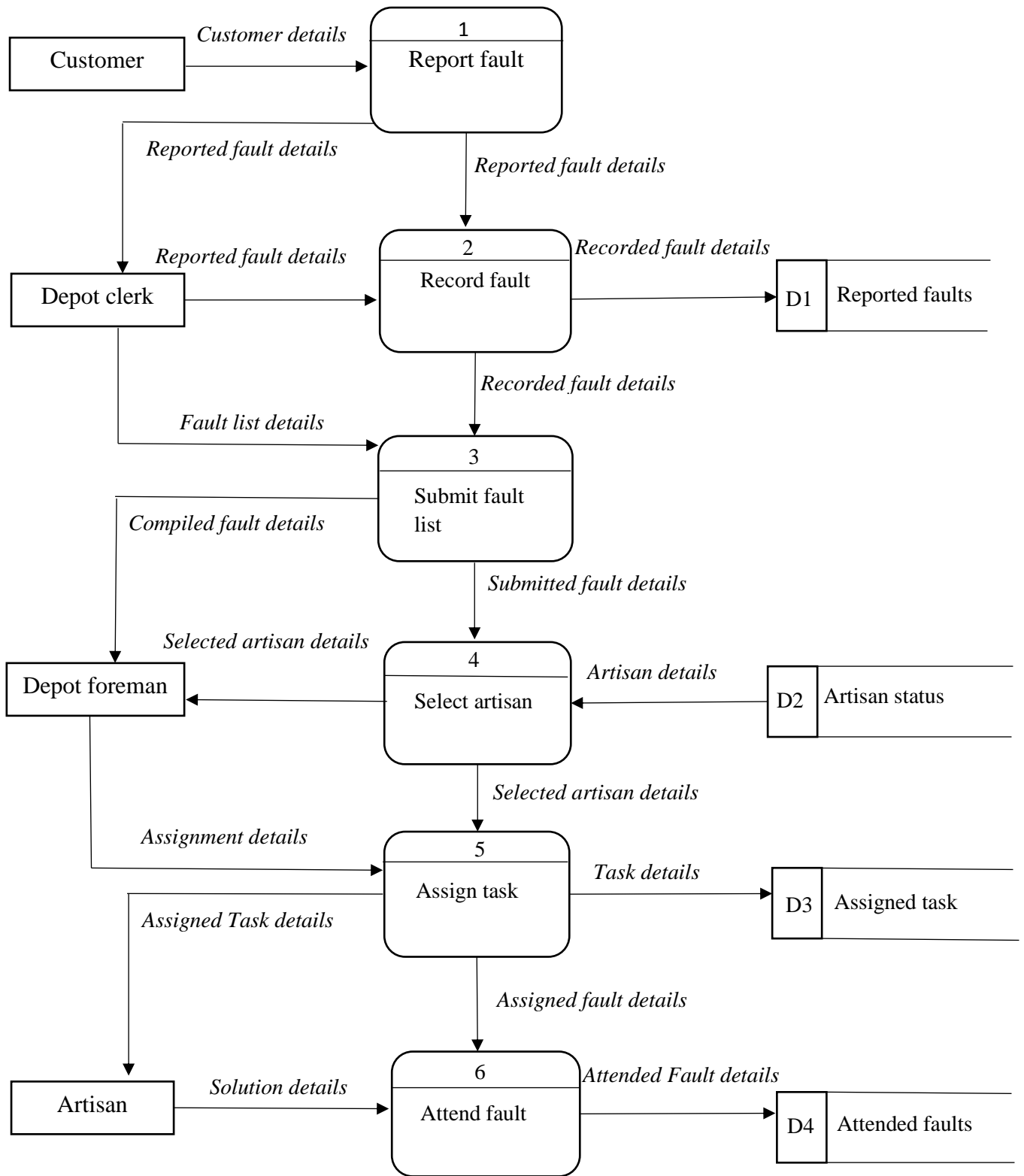
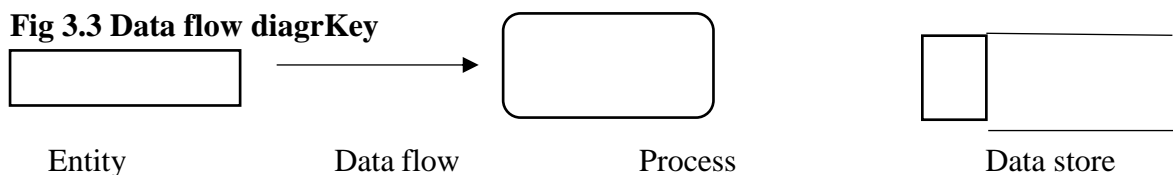


Fig 3.3 Data flow diagram



3.6 Weaknesses of the current system

- Currently there is a manual system which does not provide an ability for customers to report their faults quickly and make comments. This provides a lot of inconvenience to customers as they have to travel to the electricity company for enquiries which is costly to them.
- The current system lacks data security as files are prone theft and it does not facilitate backup.
- The current system is costly in terms of time as the retrieval of large sum of data is done manually.
- The current system does not provide the tracking of faults which tend to leave some faults unattended as it will result in customer complaints and faults are written in the book.
- There are no reports generated to see how their services are not disadvantaging others and to compute fault analysis.

3:7 Evaluation of alternatives

This is where alternatives are evaluated so that the best is chosen in order to put the correct solution in place. There are three of them which are outsourcing, in-house development and improvement that enables the proposed system to be implemented accordingly. An alternative with a low cost is being chosen as it will not lead the company to insolvency.

3:7:1 Outsourcing

It involves the buying of a ready-made software that suit the current problem faced by an organization (Butt, 2000). The company will pay license for the service as well as installation costs. The workers first need to be trained to use and maintain before implementation.

a) Advantages

- Organisation's operations are not disturbed as much of the development is done externally.
- Less costs are incurred in terms of operations and recruitments.
- Quality and efficient software being acquired if a reputable supplier is being approached.

b) Disadvantages

- It is costly to outsource.

- The software cannot be customized if there are new requirements needed to be added.

3.7.2 Improvement

This is a way of adding more features to the existing system (Hoffer et al, 2002). The features such as putting advanced security measures to the system so as to prevent potential hackers. The current system will be modified to a more efficient one so as to maximize its performance and able to satisfy its customers.

a) Advantages

- The alternative is cheap as it does not require many requirements to be added on.
- It is easy to maintain and use as well as saves time as the system already belongs to the organisation.

b) Disadvantages

- The system had a lot to add on so it's difficult to upgrade, hence a new system is required.
- There can arise some bugs which stops eve the use of the older system

The costs of repairing hardware and retraining of workers may be exorbitant

- Weaknesses of the old system are still experienced
- Some of the problems faced by the current system can only be solved by an android application such as the problem of few people accessing Zima website
- There is no room for new ideas as the improvement is done within the organisation rather than external experts.

3.7.3 In house development

This is the production of a new system software that is done within the organization taking into account all the requirements of the proposed system that meets all stakeholders (Hoffer et al, 2002). In house development enables teamwork from all stakeholders to identify their needs or requirements and help to achieve the organizational goal. The I.T will also help in gathering system requirements, do the development as well as testing it and then implement the proposed system into operation.

a) Advantages

- There is user involvement hence workers will feel the sense of being important to the organization.
- The system is easy to develop as it will be greatly supported by I.T experts, the staff and the users within the organization.
- A highly functional and quality software will be developed as the development team are within the organization responsible for collecting and analysing user requirements effectively and hence aware of the organizational culture.
- The cost of maintenance is low as developers within the organization are the once who maintains the system as they know what is required and there is proper documentation of the system.

b) Disadvantages

- There is danger of data loss if the system does deliver or work as expected.
- Development of the system will take time as compared to other alternatives.
- All risks which may be come across throughout the development and implementation will be taken by the organization.

Choice of alternative

The analysis above showed that considering in house development is the best way to develop the new proposed system as the method involves the users of the organization as they will contribute system requirements and mainly centred on the organizational problems. This is cheaper to the organization as they will be using their own resources fulfilling all user requirements rather than buying an already designed shell.

In-house development is a better method as it allows a room for maintenance or to upgrade if new requirements are needed to be added on to the system. It is also capable of integrating user needs and specifications.

Table 3.1 Alternative cost evaluation

Cost	Software Outsourcing	Improvement	Development
Development costs	1800	1300	1220
Operational costs	650	560	300
Total (USD)	2450	1860	1570

Table 3.1 shows that ZETDC should consider in-house development because the method is found to be cheaper than other alternatives. This method is cost effective because much of the staff within the organization are skilled hence they will be a quality software to be produced.

3.8 Requirement analyses

Shelly and Rosenblatt (2010) define requirement analyses as the process of determining user expectations for a new or modified product for a particular organisation. It also defines what the system does. It also explores what the system does and its processes for it to operate effectively. Requirements analysis therefore describes the proposed system's intended behavior as well as its associated constraints. Requirements are divided into functional and non-functional requirements. Functional requirements are tasks the system must support, whilst non-functional requirements are constraints on various attributes of these tasks.

3.8.1 Functional requirements

Functional requirements describe what proposed system should do, all the activities involved within the system and can also be referred to as product features (Duffy, 2011). Moreover, Somerville (2008) argues that the basic functions that should be provided by a system should be specified as functional requirements. There will be a backend and a front end. Therefore, the ZETDC fault reporting and tracking mobile application is expected to provide the following functions:

3.8.1.1 Front end for clients

- **Fault reporting**
It enables clients to report their faults on the application. When reporting a fault, a client will go on the application and post the fault and putting the address where the fault occurs, then the system will automatically reply giving him feedback that will be there shortly.
- **Customer service management**
The system allows the customers to post their views, queries and comments through the web platform and they will be provided with the feedback. As a result of this good loyalty to the organization will be gained as customers complains will be addressed to easily.

3.8.1.2 Back end for the organization.

- The depot foreman must be able to track if faults are being resolved on the application using track code. Thus, the artisans have to log on to the system and record the faults resolved so that the depot foreman has to track all faults so as to be able to make reports at the end of the month so see which materials are redundant that causes problems in a certain area.
- All client's activities such as giving responses to queries, updating etc. are managed by the administrator.
- The depot foreman should be able to allocate tasks to artisans.
- It will be easy for managers to make decisions as the system will be generating and producing monthly reports.

3.8.1.3 Use Case diagram

The use case diagrams highlight the activities of the system from a user's point of view. The use case diagram shows the interaction that exists between different roles and different functions of the system. It is referred to as the upper level view of the system, valuable exclusively when presenting to different stakeholders. Use case comprises of 4 objects which are the actor, use case as well as the system and association. **Fig 4.3** shows a use case diagram for the proposed system.

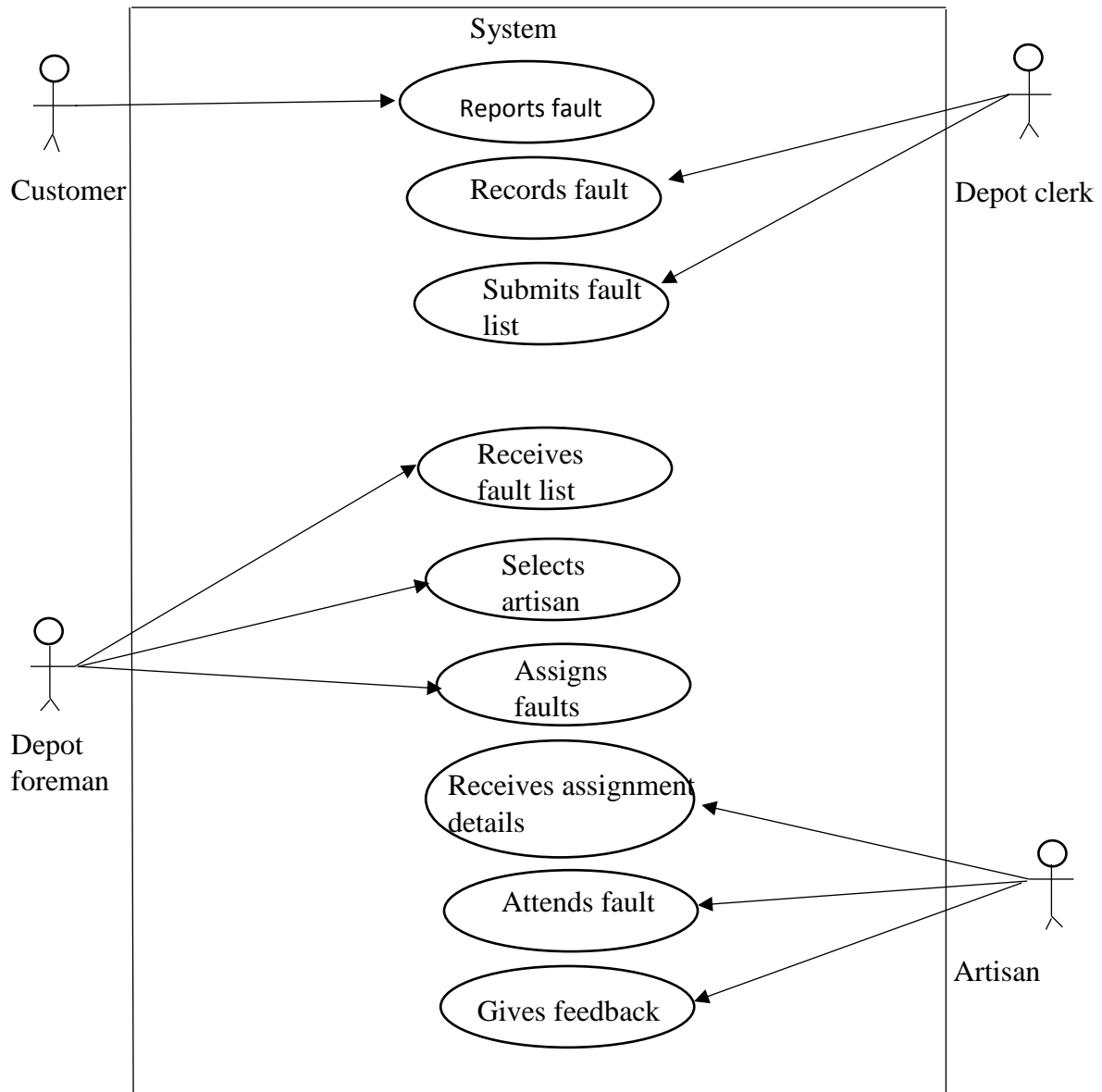


Fig 3.4 Case diagram

3.8.2 Non- functional requirements

According to Marakas and O'Brien (2013), non-functional requirements describes how well the software does and also the quality aspects of the system as well as the how the proposed system should deliver in terms of basic functions of the system. The non-functional requirements should also include constrains of the system.

a) User-friendly interface

- The system should fast, simple to understand and use.
- It will have a comprehensible user interface.

b) Security Issues

The system should ensure security through:

- Varying access levels for different system users
- Securing authentication over use of secret codes and admission rights.
- Making sure that passwords comprises of a minimum of 6 characters.

c) System efficiency and output

The system should:

- Facilitates quick retrieval of information whenever needed.
- Less response time and reduction in operational costs.

d) Compatibility

The system should be:

- Adapt to various applications within the organization.
- Suit with other operating systems.

f) Back up

The system should be:

- Minimize data loss after a system failure or crush
- The database should be regularly backed up at least thrice a day to enable consistence.
- The system and its database should be maintained and controlled by the administrator to ensure data integrity.

3.9 Conclusion

The phase managed to assess the possible alternatives to effectively solve the problems and it was found that in-house development is the most appropriate technique because it was found cheap and functional and non-functional requirements were also identified. New system requirements and specification were determined through various data gathering techniques. The next chapter will be looking at the actual design of the proposed system which involves architectural, physical, database, program, interface and security design.

CHAPTER 4: DESIGN PHASE

4.1 Introduction

The main emphasis in this phase is the architectural design, interface design as well as database design of the new system. This is how the system is going to be developed considering its functional and non-functional requirements. The functional requirements are used to design the processes that the proposed system should execute and the non-functional requirements are used to design the system's actual interfaces that will be user friendly and appear good to the users. It also involves activities and process analysis of the proposed system.

4.2 System Design

This process involves merging different system modules, elements and also system interfaces so as to fulfil the system requirements of the proposed system. It also shows the entire system architecture which involve the software, hardware and how they interact with each other. The proposed system will have the following features

- Efficiency – The user should perform various task on the system and the processing time should be short. The system should provide quick feedback to the user and provide efficiency in recovery if any fault arises within the system.
- User friendly - The system must provide a friendly way for users to easily use the interface and reduces user's consultation on using the proposed system.
- Security – The system should be in a position to restrict access to unauthorized users and prevent loss of confidential information. Data should be always kept safe, procedures and policies should be put in place to deal with security breach.
- Maintainability – The proposed system must be maintained easily and allows room for improvement if any changes are introduced.
- Effectiveness – It should perform as intended and as well as producing the required results.

4.2.1 How Fault Reporting and Tracking Android application works

The Fault reporting and tracking android application will enable clients to report faults, make comments on the application and provide feedback to clients.

The system also provides valuable information thus notices to its valuable clients on the application and also tracking of faults by the management. Tracking of faults helps the management to draw up reports that analyses fault status of attended and unattended faults.

4.2.2 Context diagram of the proposed system

The diagram show the entities and their relationship with the system that exist in the overall process (Shelly and Rosenblatt, 2010). It consists of many entities and single process and the context diagram of the current ZETDC fault reporting and tracking reflects interactions between clients, depot clerk, depot foreman and artisan and this is shown on fig 4.1

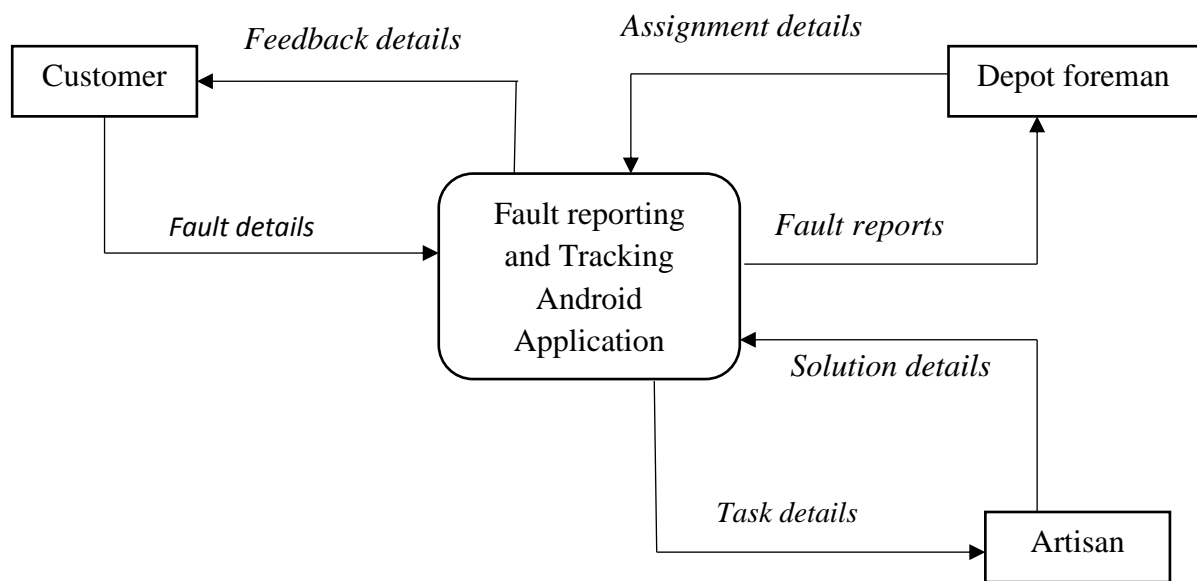
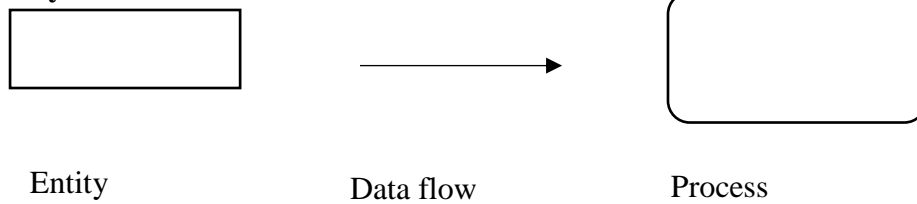


Fig 4.1 Context diagram

Key



4.2.3 Data flow diagram of the proposed system

It is a diagram that designed to show how information flow from one entity to another and the processes that are involved within the system. It uses a set of tools which are data stores, entity, processes and data flows to clearly represent the operations performed within the system. This is illustrated by the diagram in fig 4.2

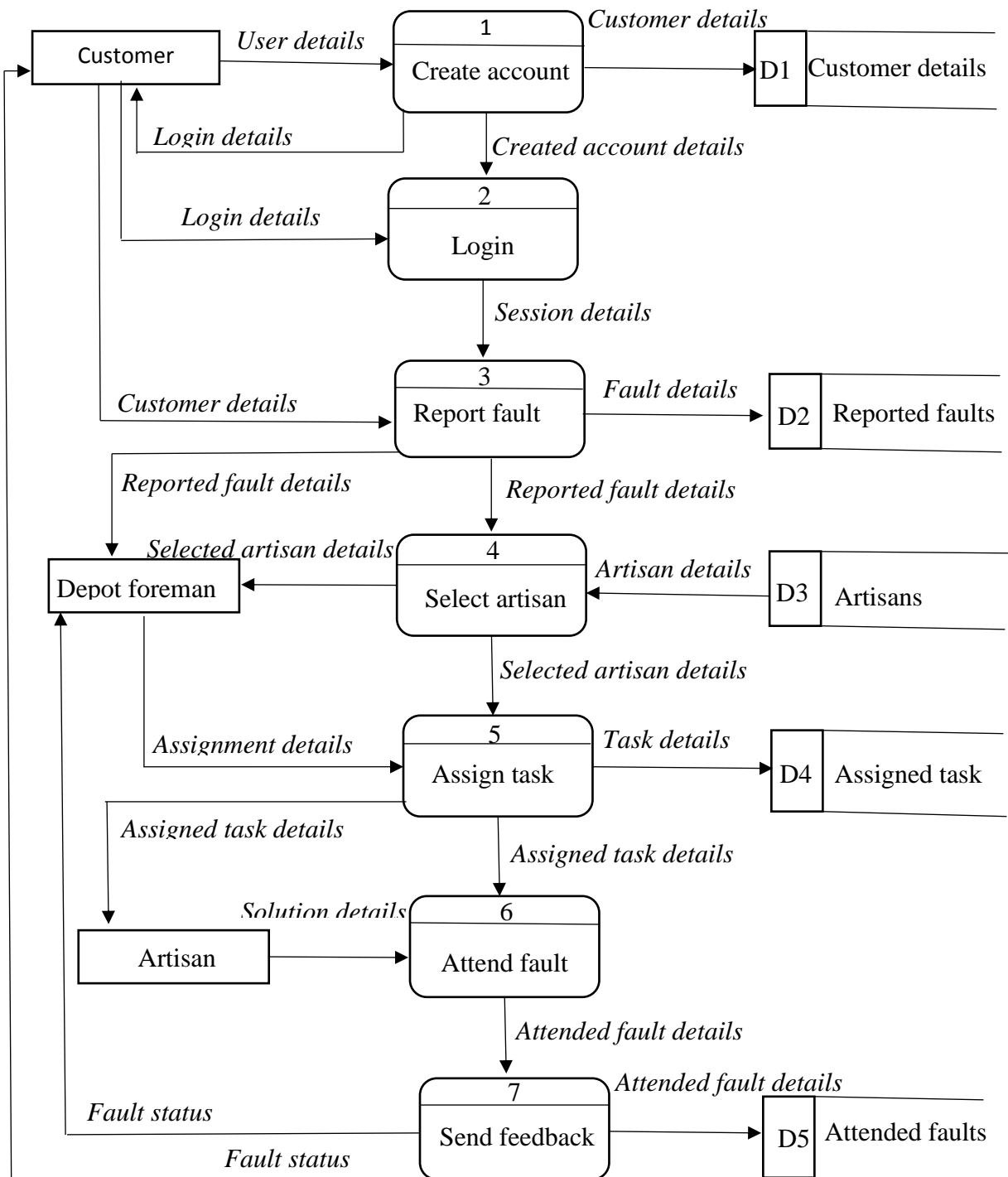


Fig 4.2 Data flow diagram

Key



Entity



Data flow



Process



Data store

4.3 Architectural design

This is the process of presenting data modules and structures that are required for developing the proposed system (Somerville, 2008). Architectural design describes the hardware and software required to suit the available requirements. It is also called high-level design with various subsystems. It describes fully the platform on which the system will run. Client server architecture is the one used on the development of the proposed system as client will manage user interfaces. The interfaces will include data entry, reporting, querying and the server will handle data management such as storage, access to it. This is a useful architecture as it permits customers to send their queries and request to the database server. The database server will then processes the requests and provide the response to the customer via client/server interaction.

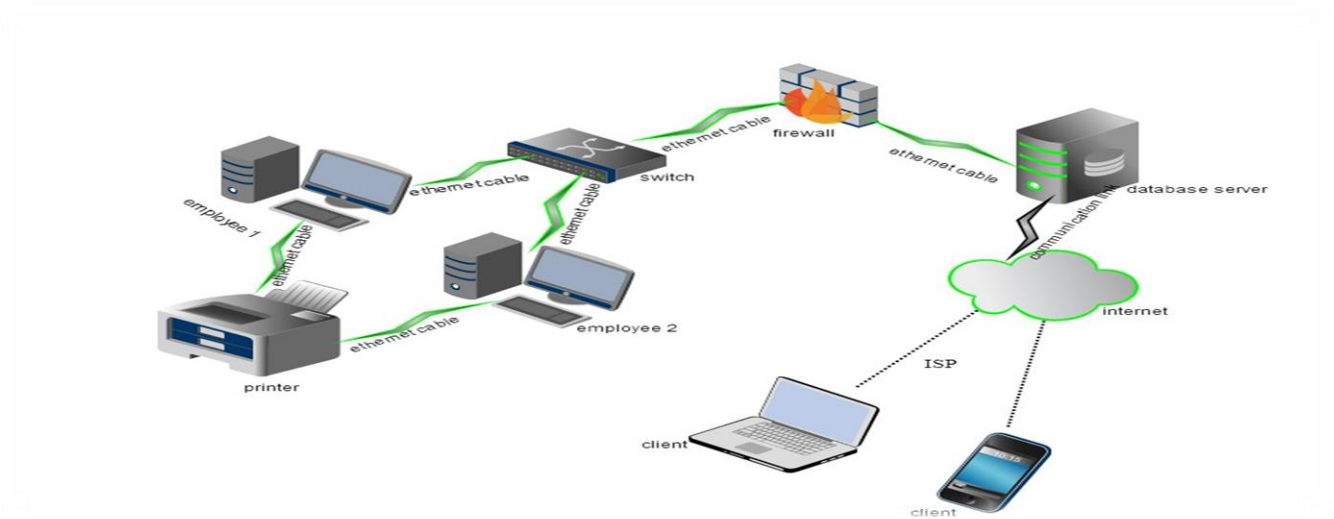


Fig 4.3 Architectural design

4.4 Physical design

It is the interaction of the hardware and software components via the network within the organization (Coronel and Crockett, 2008). The system is going to be accessed via the network and it is connected to Internet. It also enhances the translation of logical model of the system to the physical model of the system and how they are integrated to the database server. ZETDC customers accesses the system through the Internet using their mobile devices to communicate their issues to the organization reporting electricity faults. Then ZETDC depot foreman will access the system through SAP on the local area network. System inputs and outputs processes are to be considered and the verification of data is also done so as to produce the required output. This is illustrated in fig 4.4

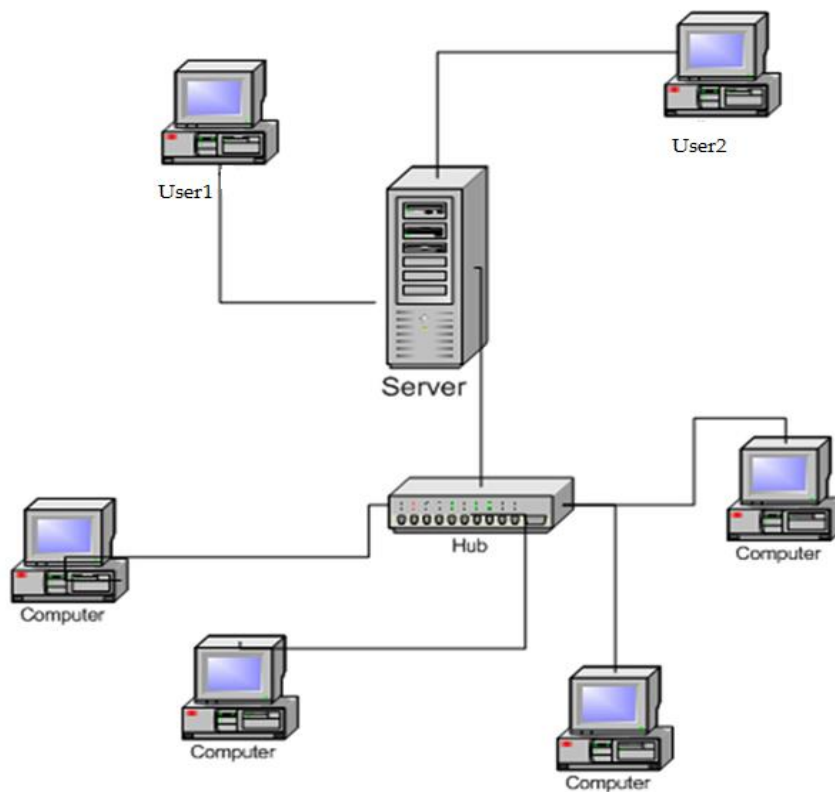


Fig 4.4 Physical design

4.5 Database Design

Somerville (2008) defines database design as the process of developing an engineered mechanism that facilitates data storage as well as enabling ease access of information. A well-designed database should provide data integrity, data consistency as well as minimal data redundancy. Data security should be enforced to prevent unauthorised users. Database design also involves the database architecture of the system. The ANSI SPARC architecture will be used in developing the new system. It helps users to represent data in an understandable way and it has got three different layers shown in fig 4.5

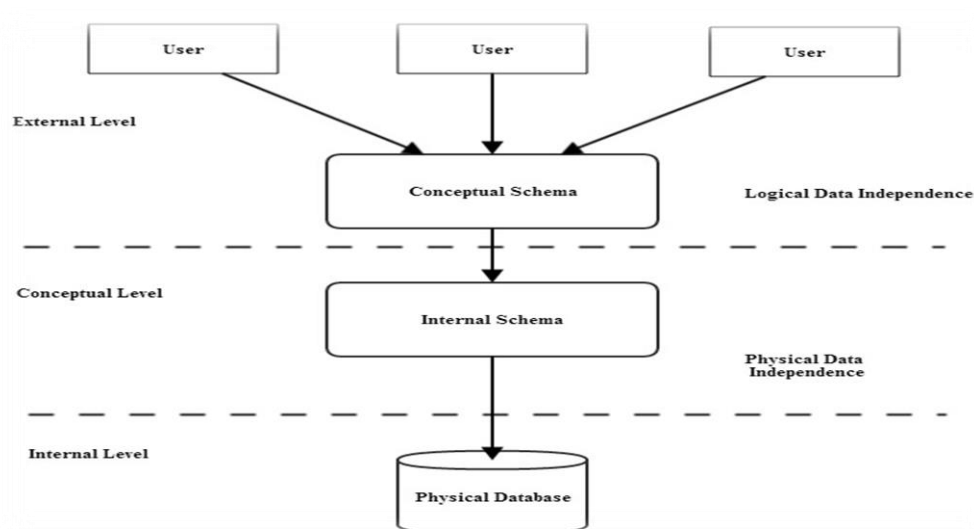


Fig 4.5: Database Design (Somerville, 2008)

4.5.1.1 External level

It is also known as the view level because it shows how a user views the database. This level allows a user to access a modified form of data in the database. It hides the working of the database from users. It maintains the security of the database by giving users access only to the data which they need at a particular time (Somerville, 2008).

4.5.1.2 Conceptual level

This level represents the community view thus it shows how the database is structured logically. It also shows the relationship between the data members of the database, exactly what data is stored in it and what a user will need to use the database (Coronel and Crockett, 2008). It helps hide the complexity of the database and hides how the data is physically

stored in it. It provides a global view of the database, as well as the hardware and software necessary for running it.

4.5.1.3 Internal level

This level deals with how the stored data on the database is represented to the user. This level shows exactly how the data is stored and organized for access on your system (Somerville, 2008). This is the most technical of the three levels. It also include figuring out the right space allocation techniques, data compression techniques ,security and encryption and the access paths the software can take to retrieve the data.

4:5:2 Logical design

These are collections of related data held in a structured format within a database. It consists of columns, and rows. Zimbabwe Electricity Transmission and Distribution Company will access the tables shown in the following figs.

Table 4.1: Login Table

Field	Type	Descriptions
First Name	Varchar	Administrator's first name
Last Name	Varchar	Administrator's last name
ID	Varchar	Administrator's ID
Email	Varchar	Administrator's email
Password	Varchar	Administrator's password

Table 4.2 Fault reporting table

Field	Type	Descriptions
First Name	Varchar	Customer's first name
Last Name	Varchar	Customer's last name
Fault	Varchar	Customer reporting fault
Email	Varchar	Customer's email
Address	Varchar	Customer's address
Date	Date	Customer's date of reporting fault
Phone number	Varchar	Customer's phone number

Table 4.3 Pending faults table

Field	Type	Descriptions
Date	Date	Date where the fault was reported
Reason for not attending fault	Varchar	Explanation on failure to attend a fault
Address	Varchar	Address of fault premise
Customer Name	Varchar	Customer's name
Customer phone number	Varchar	Customer's phone number
Artisan	Varchar	Artisan who supposed to attend the fault
Foreman	Varchar	Depot foreman who assigns artisans

Table 4.4 Resolved faults table

Field	Type	Descriptions
Date	Date	Date when the fault was resolved
Fault status	Varchar	Updating the faults status: Resolved
Address	Varchar	Address of the house
Artisan	Varchar	The one attended the fault

4.5.3 Enhanced Entity Relationship diagram

Kendall and Kendall (2005) stated that an enhanced entity relationship diagram shows different data structures that exist in the system and how the data flows in the proposed system. It is shown in fig 4.6

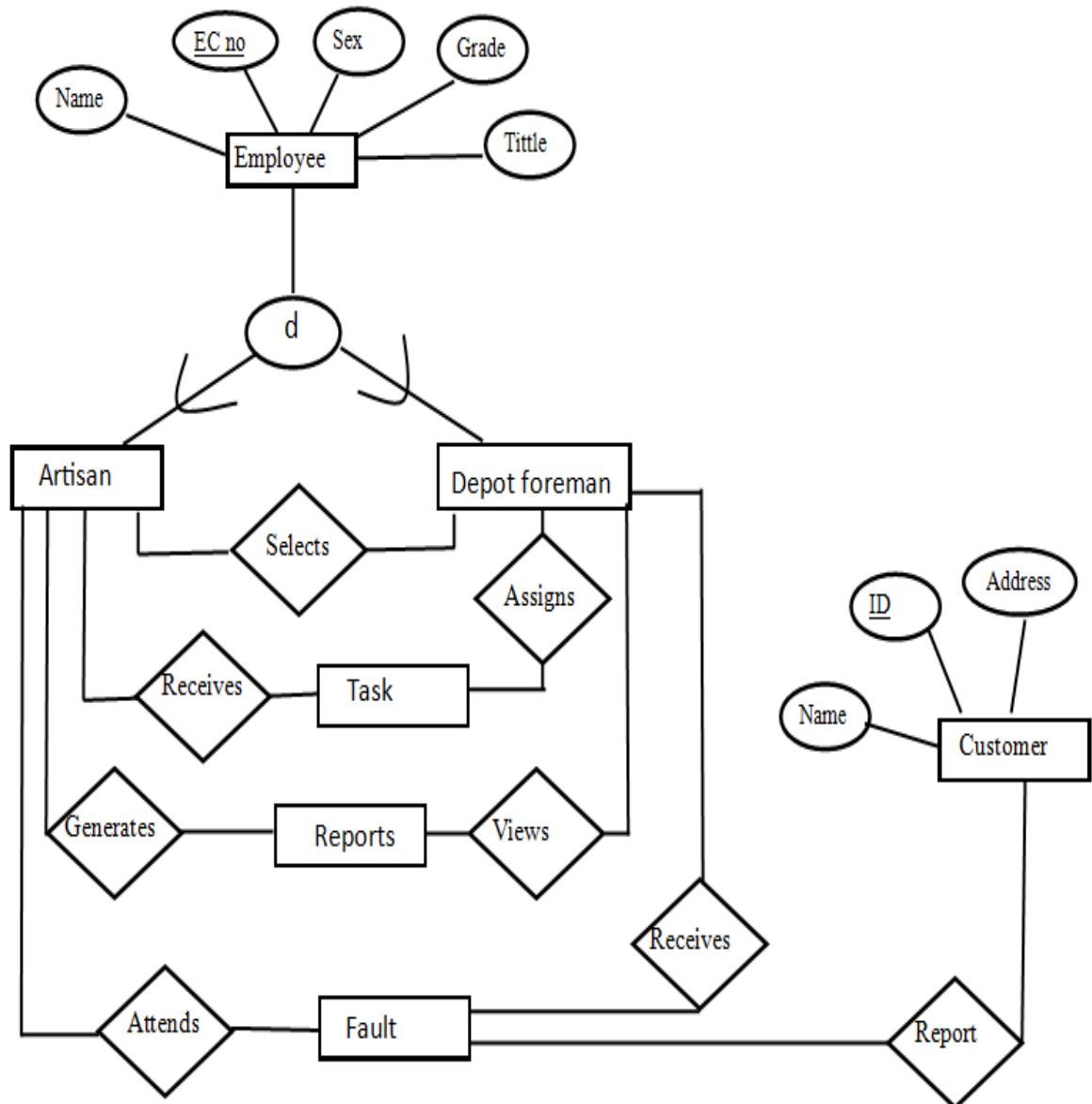


Fig 4.6 Enhanced Entity Relationship diagrams

4.6 Program design

Dennis (2012) defines program design as a method for designing and documenting methods and procedures in software. It also describes as how the various pieces of code will be integrated to form units, modules and programs. It also ensures that code is understood, organized and integrated. Program design of the proposed android application can be illustrated through the use of package, class and sequence diagrams.

4.6.1 Package diagram

A group of logically connected components thus expressing the interaction of the software modules which are usually in classes, to form packages (Somerville, 2008). The logical relationships amongst packages are shown in fig 4.7

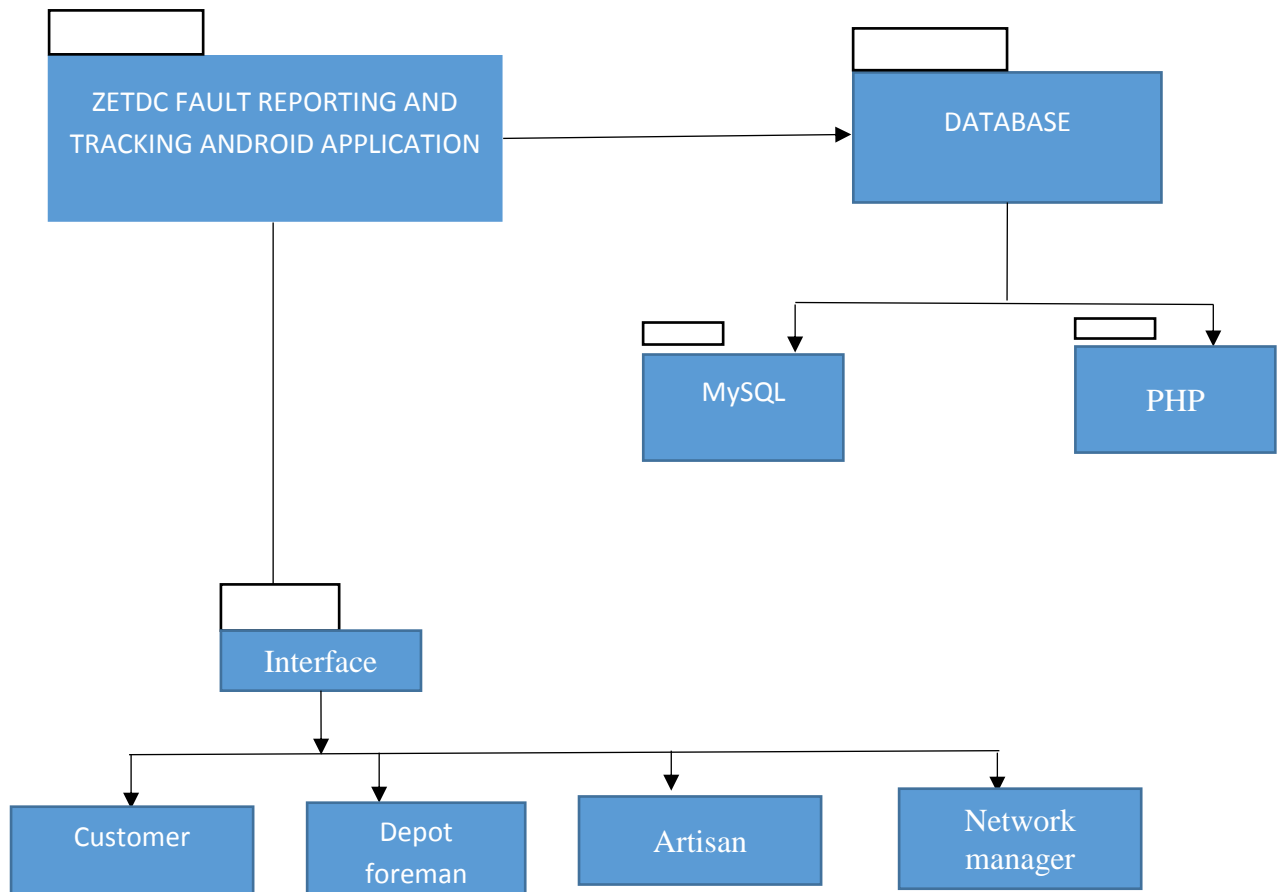


Fig 4.7 Package diagram

4.6.2 Class diagram

George and Valacich (2002) define a class diagram as unified modelling tool used in object oriented development to show the relationship between objects. It also shows the data structures and the operations that are performed on the data. The class diagram includes classes with attributes and behaviours as shown in **fig 4.8**

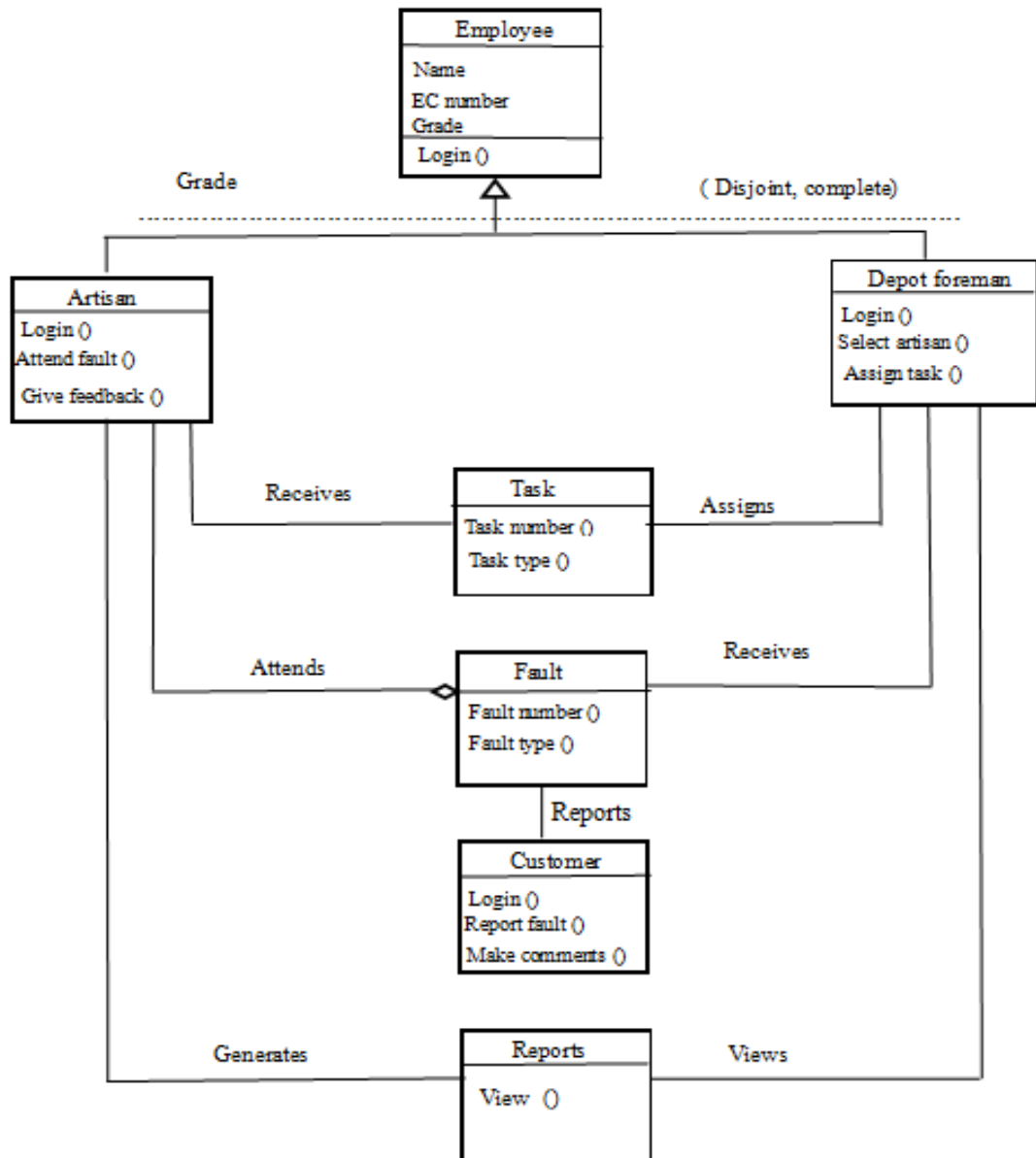


Fig 4.8 Class diagram

4.6.3 Sequence diagram

Aggarwal and Singh (2009) define a sequence diagram as an interaction diagram that shows how objects operate with one another and in what order. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. They also show the object's lifelines and interactions in time sequences. This is shown in **fig 4.9**

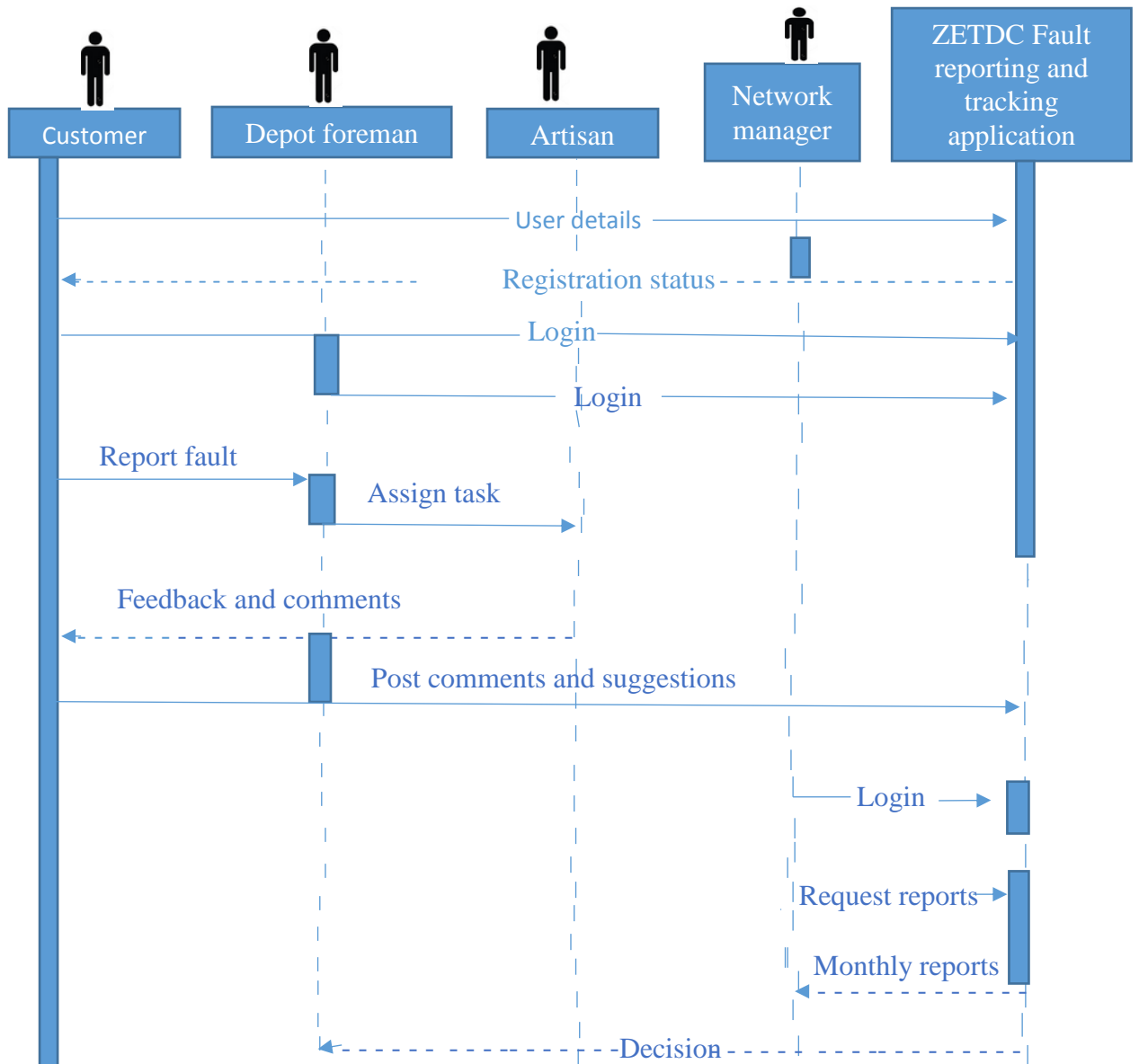
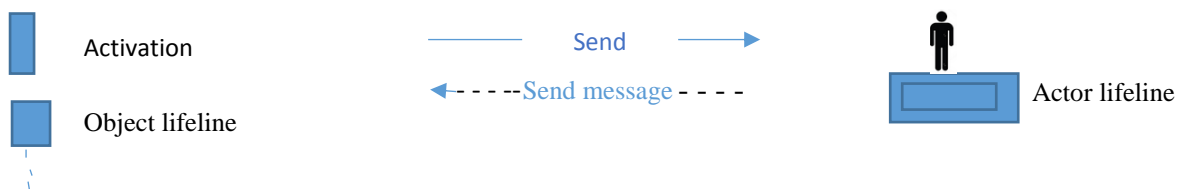


Fig 4.9: Sequence diagram

Key



4.7 Pseudo code

Login

Customer login

Accept email and password

Select email and password *from* customer table

If records match

Direct customer to home page

Else display login failed message

Adding a new user

Accept email and password

Select email *from* user table

If email does not exists then

Register user

Else

Display appropriate error message

Report fault

Enter fault details

Enter contact details

If fault status changed to attended fault then

Then

Fault has been resolved

Else

Fault not resolved

Fault Report Generation

Compare number of reported faults with number of resolved faults

If the number of reported and resolved are equal

Then

Produce a satisfactory report output

Else

Error message: Revisit unresolved faults and management should take proper action

4.8 Interface design

The interface design is the one that allows designers to design variety of interfaces which includes forms, reports and screen displays (Dennis, 2012). The design of the proposed system was designed in such a way that it will be easy for user to interact with the system easily. Also to reduce the amount of time trying to get how to access the services provided by the system and able to accomplish the task within a short period of time.

4.8.1 Menu design

The menu design allows the user to navigate around the system to choose the desired option (Dennis, 2012). It enables users to perform various operations through navigating system functions.

4.8.1.1 Main menu

The main menu will consist of sub menus that supports the customer and the users within the organization. The users will access the main menu after they log in the system and then it will appear. The respective main menus are shown on the **fig 4.10**

HOME

WELCOME TO ZETDC

Fault Reporting and Tracking Application

Enter username and password

Username

Password

[Sign in](#) [Sign up](#)

[Forget password](#)

Fig 4.10 Main menu

4.8.1.2 Sub menus

Submenus provides the user with more options to execute system processes. The links for carrying basic database operations such as inserting, updating and deleting records are usually located on sub menus. They enable users to carry out various operations within the system modules thus in the front and back end. These will be shown in **fig 4.11**

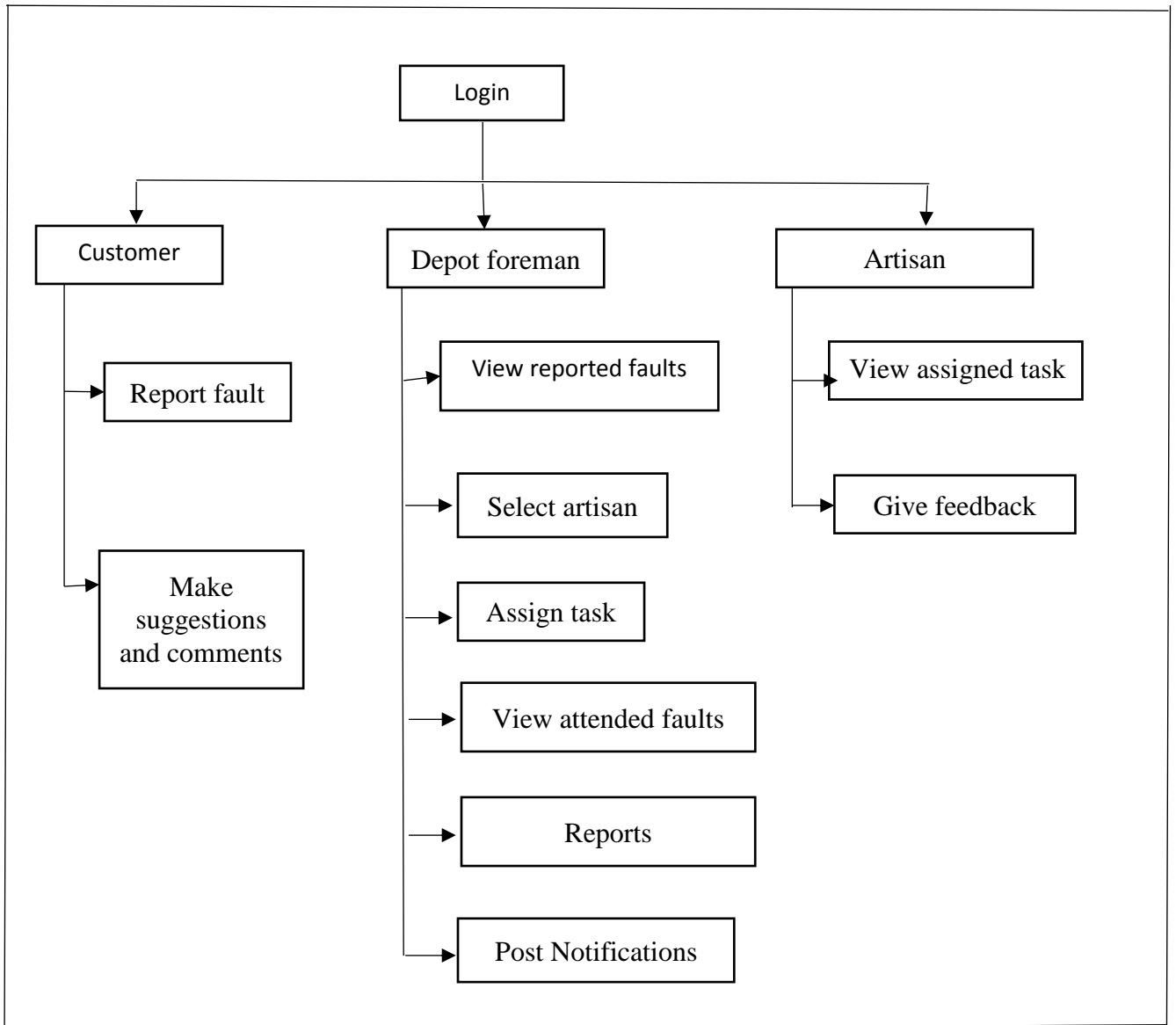
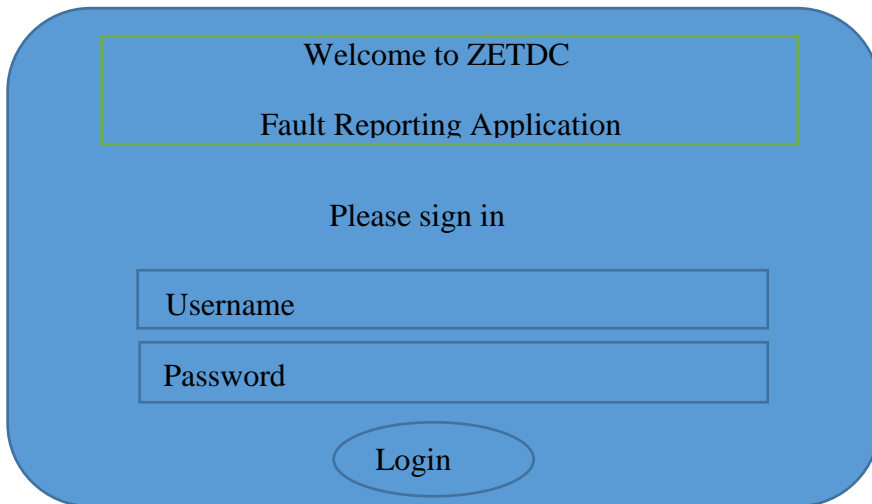


Fig 4.11: Sub menu

4.8.2 Input design

Shelly and Rosenblatt (2012) define input design as the mechanism that allow users to enter data in the system. This data will be quickly processed into computer readable format to produce the required output. The format of the data on the entry point should match the format of the data that can be held by the fields on the database table (Lowery, 2002).

Fig 4.12 shows the log in page for users of the system. The user must input username and password to get access to the system.



Welcome to ZETDC
Fault Reporting Application

Please sign in

Username

Password

Login

Fig 4.12 Log in

Fig 4.13 shows the sign up page that allows users to create their profile so as to get access to the system. The username and password created will be used to login to the system.



Welcome to ZETDC
Fault Reporting Application

Please sign up

Name

Surname

ID

Email

New password

Confirm password

Submit

Fig 4.13 Creating user account

Fig 4.14 shows the page that is going to be accessed by the customer when reporting fault.

Welcome to ZETDC
Fault Reporting Application

Name

Address

Report fault

Phone number

Attach picture

Submit

Fig 4.14 Report fault

Fig 4.15 shows the page that is going to be accessed by the depot foreman when assigning a task to the artisan.

Welcome to ZETDC
Fault Reporting Application

Artisan name

Address

Fault description

Submit

Fig 4.15 Assign task

Fig 4.16 shows the page that is going to be accessed by the artisan when giving feedback to the depot foreman about the fault status.

The image shows a blue rounded rectangular interface for a 'Fault Reporting Application'. At the top, it says 'Welcome to ZETDC' and 'Fault Reporting Application'. Below that is the text 'Fault feedback'. There is a large rectangular input field labeled 'Enter description'. At the bottom center, there is an oval button labeled 'Update fault'.

Fig 4.16 Feedback

4.8.3 Output design

The mechanism that delivers the processed information to the users by providing quick query responses (Somerville, 2008) The major role of output designs is to deliver the outcomes of the request made from the system by the system users (Shelly and Rosenblatt, 2010). The system's usefulness is evaluated by considering the outputs of the system and these output generation is the major reason for developing a system. Lowery (2002) stated that the system should produce valid and accurate output and release relevant output at the right time.

Fig 4.17 shows the page that is going to be accessed by the network manager when analysing monthly.

Welcome to ZETDC							
Fault Reporting Application							
Name	Surname	Address	Date attended	Fault description	Fault status	Artisan attended the fault	Comments
Tatenda	Chikosi	9 Dorchester Fitchlea, Kwekwe	16-02-17	Power lines struck by lightning	Resolved	T.Charumbira	
Max	Giyane	34 Pinegroove , Newtown	21-03-17	Transformer fault	Not resolved	T. Magaso	No transformer in stock

Fig 4.17 Faults

Fig 4.18 shows the page that is going to be accessed by the customer after reporting the fault.

Welcome to ZETDC	
Fault Reporting Application	
The fault was noted...Our guys are coming shortly.	
Thank you.	

Fig 4.18 Notification

Fig 4.19 shows the page that is going to be accessed by the customer after the fault has been attended.



Fig 4.19 Fault status feedback

4.9 Security design

Zacker (2000) defined security design as the act or procedure of designing and implementing measures, with the objective of reducing vulnerabilities and risk that might affect the operation of the proposed system to both hardware and software tools. Security system is a vital component as it prevents the system from various threats. However policies and procedures should be designed to safeguard the new developed system from physical and non-physical damages that may hinder its performance.

4.9.1 Physical security

Hughes and Cottrell (2009) define physical security as the prevention of tangible assets of an organization such as property from being harmed, natural disasters or terrorist attacks and unauthorized access to sensitive data within the computers. It also focuses in ensuring a safe physical environment for an information system.

Some measures that can be employed to aid physical security include using a number layers of reliant frameworks which include closed circuit television (CCTV) surveillance, security

monitors, secured access control conventions and protocols. However, backups will be done daily and the information will be stored in separate server as well as in the cloud.

4.9.2 Network security

The protection of computer network and all of its traffic and resources by an organisation (Kendall and Kendall, 2005). It also involves the protection of data during the exchange. Network security includes the process of physical and software preventive measures to protect the network infrastructure from unauthorised access. Some measures aid to the security of a network include the use of an antivirus software, encryption of data, authentication and malware detection. Firewalls are also used to safeguard the server from Internet vulnerabilities.

4.9.3 Operational security

Kendall and Kendall (2005) defines operational security as vulnerabilities that are likely to cause harm to the proposed system during its operational life time. The operational threats can be within or outside the organization. Some of the threats are difficult to control such as fire. Hackers are also one of the major threats to the system as some of the user will exposing some of the sensitive information. Some ways to assure operational security can be through the use of email encryption during communication to counter spoofing and electronic eavesdropping. The use of firewalls, antivirus protection software and access login software are the measures put forward by the developer to ensure operational security of the information system.

4.10 Conclusion

The chapter has managed to show the database schemas that are to be used when developing the proposed system. It also directs the developers on how the system is going to be assembled. This phase managed to discuss the software design part and a good graphical user interface was designed which is user friendly. This phase has managed to discuss the design of the system such as interface physical and architectural design of the proposed system. The next chapter will be dealing with the implementation and maintenance of the proposed system.

CHAPTER 5: IMPLEMENTATION PHASE

5.1 Introduction

The implementation phase is mainly focused on the code, installation, testing and maintenance of the proposed system. It also involves the process of implementing the developed system into operational environment so as to test it and identify loop holes in the earlier stages. This phase also determines whether the developed system's objectives have suit to its operational environment to verify if the system is error free. Various changeover techniques are discussed and the most suitable technique is chosen and used. It also include the various maintenance plans to assess the system performance as well as its adaptability to the environment.

5.2 Coding

Pressman (2005) defines coding as the process of writing of instructions for the computer to execute. The written code is then tested to make sure that it performs as expected. According to Butt (1994) also defined coding is a technique of designing a program that can be executed to come up with a functional system. For the development of ZETDC Fault reporting and tracking android application it incorporated PHP, Java script and Android Studio.

5.3 Testing

Laudon and Laudon (2013) define testing as a process of executing an application program or a system with the intent to discover software bugs that affects the operational life of the system. This is a useful process in software development because it helps the developers to quickly bugs in the early stages of system implementation. However a quality and an efficient system will be produced which will satisfy the users of the system. In addition, Somerville (2007) asserts that testing is the act of validating and verifying that the software program meets the business and technical requirements that guided its design and development. Testing is broken down into various modules that the system go through in each module. The testing process is shown in fig 5.1

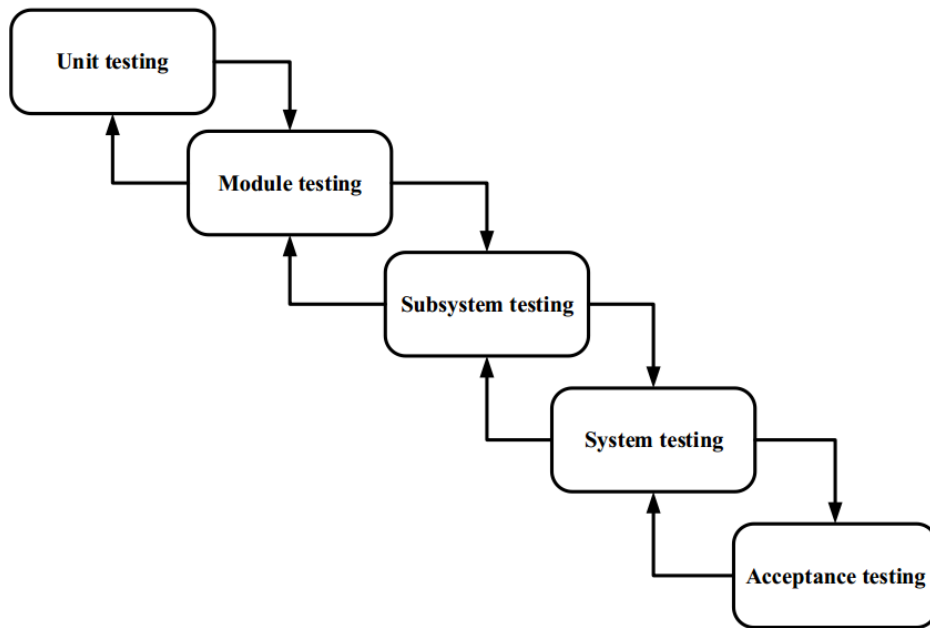


Fig 5.1 Testing Process (Somerville, 2007)

5.3.1 Unit testing

Whitten, Bentley and Dittman (2004) define unit testing as the process of testing of various individual software components or modules of a particular system. This is usually done to ensure that modules of the system are performing as intended and they are examined independently. This act is typically done by programmers and not testers as it requires detailed knowledge of the internal program design and code (Kendall and Kendall, 2005). Unit testing can be done in two variation thus white box and black box testing.

5.3.1.1 Black box testing

Hudges and Cotterell (2009) define black box testing as tests that are based on requirements and functionality of the application program. This aspect of black box testing is a valuable type of test that exposes hidden bugs or error in the code snippet and ensure a thorough test on the implementation.

5.3.1.2 White box testing

Whitten et al. (2004) define white box testing as tests that are based on knowledge of the internal logic of an application's code. Tests in this concept are based on coverage of code statements and conditions.

White box testing is manly centred on code structure of the application rather than its functionality. This helps in quick detection of errors or bugs in the code.

5.3.2 Module testing

Jawadekar (2004) defines module testing as testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, and client and server applications on a network. This type of testing is especially relevant to client/server and distributed systems. It also involves the breaking down of the system into modules for example login, create user account or sending a query to the system. This is done repetitively till all modules are effectively combined until there are no errors found in the code.

5.3.3 Subsystem testing

Kendall and Kendall (2011) indicated that a sub system test gathers all the modules which will be integrated and tested as a sub-system and more intact application. By sub system testing the analyst is able to test for module compatibility and functionality. These are combined to yield a real information system if all tests are done. This test is responsible for discovering hidden errors which occur within the integrated system modules.

5.3.4 System testing

Kendall and Kendall (2011) define system testing as the process of testing the entire or the complete system to ensure that it meets the specified requirements or the system objectives. The main purpose of system testing is to assess the system's compliance with the stated requirements. The errors or bugs of the entire system will be discovered as a hole through this type of tests. It also aims at finding out the effectiveness of the new system as compare to the old system. This will be done by entering data into the system and produce the output must be produced as expected. System testing is essential as it enables the testers see if they are any errors that might occur whilst the system is in operation.

5.3.5 Acceptance testing

Whitten et .al (2004) define acceptance testing as a type of test that verifies if system meets the customer specified requirements. This is usually done by users of the system to determine whether to accept application or not. The feedback that is provided by the users are the ones that are used to perfect the final system thus users are key actors in software development. It consist of two components thus alpha and beta testing.

5.3.5.1 Alpha testing

Alpha testing is defined as the process of testing the system at the end of development and minor changes will be done after such test has occurred (Rosenblatt, 2011). The changes such

as spelling mistakes, broken links will be tested. If errors are found during the process, the developer will be able to provide solutions to the problems.

5.3.5.2 Beta testing

O'Brien (2005) defines beta testing as a type of test that is done by end-users. This is also the final testing before releasing the application for commercial purposes. The feedback that will be provided will be used to develop a full system after user end user testing. This is a vital aspect as it helps the developers to come up with a quality software that is error free.

5.3.6 Testing strategies

Various testing strategies were enforced to verify and ensure that the developed system works efficiently and effectively. There are errors such as syntax errors and database connection errors that were noted and these were rectified easily.

5.3.6.1 Validation

Jawadekar (2004) indicates that validation is the process of evaluating the final product to check whether the software meets the business needs. This involves functional, regression and system testing. This is a useful aspect in system implementation as it determines whether the application package is fit for use and satisfies the business needs. It is also the process of checking whether we are developing the right product or not (O'Brien, 2005).

a) Test Cases

Test cases are tests used to determine if the system is able to produce quality output as intended. They are carried out using both invalid and valid data.

The admin at the backend is required to enter user details when creating a particular staff member as to gain authorisation to the system but if the admin tries to create a user without entering data in the specified fields a pop up message will show.

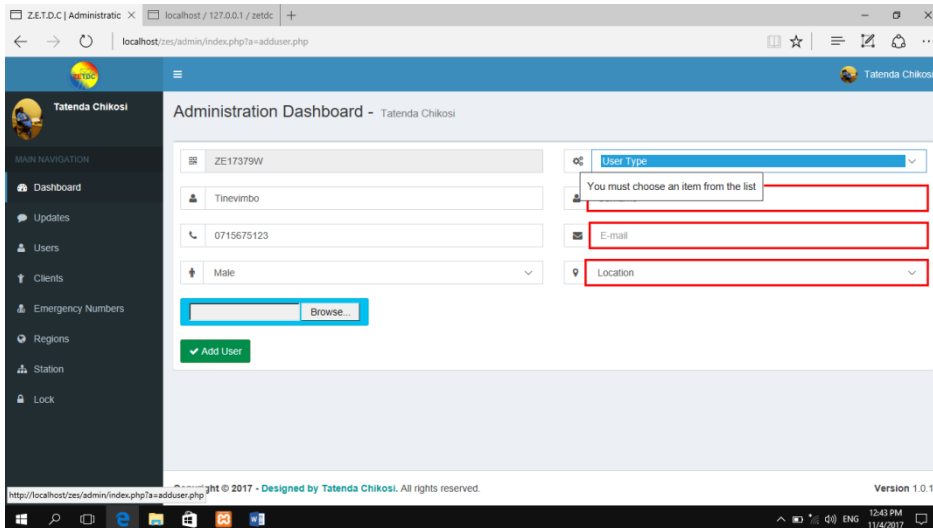


Fig 5.2 User creation validation

If the user enters wrong credentials upon login then an error message will appear.



Fig 5.3 Login validation

Creating user account

The admin is the ones who create staff accounts and Fig 5.4 shows a page where an invalid email is entered when creating a user. An error message pop ups.

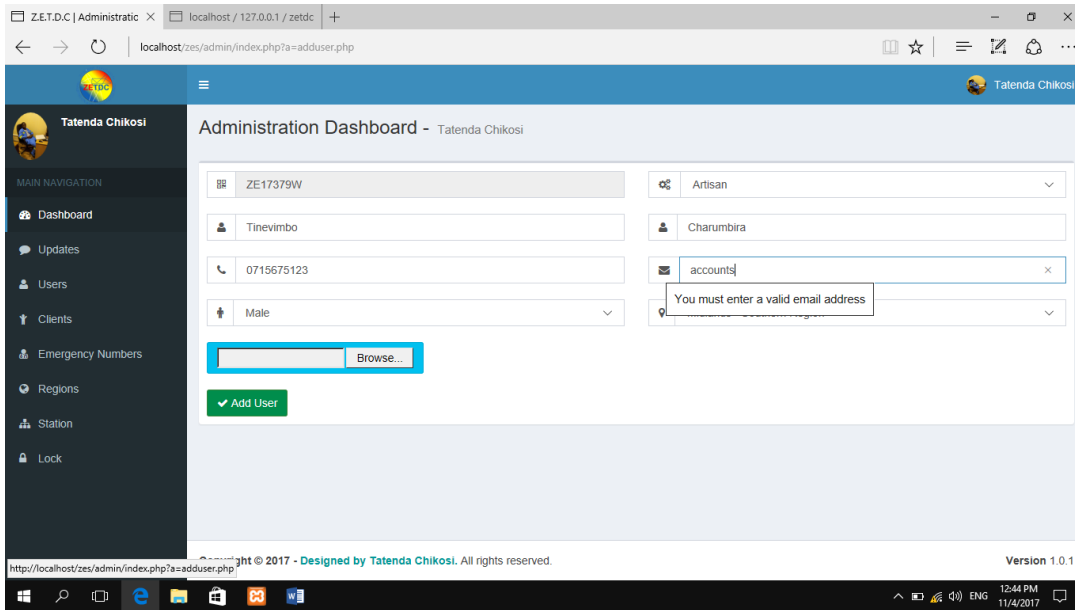


Fig 5.4 Creating user account
Tracing of faults

This is when a wrong fault number is entered in the system and an error message will appear.

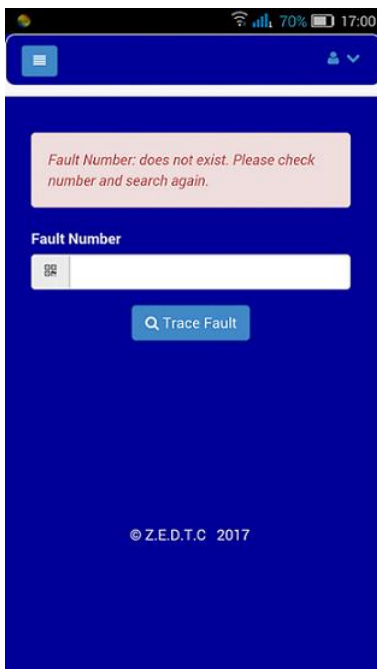


Fig 5.5 Tracing of fault
b) Security Testing

Grady (1997) argues that security testing as procedures that are taken to ensure efficient security to the system thus preventing unauthorised user to gain access to the system. It ensures data integrity to the organisational data. The android application will be having the

use of user access levels, authentication and authorisation of usernames and passwords. Virus protection will also be enforced.

User access levels

A general client can only see information relating to his or her account.

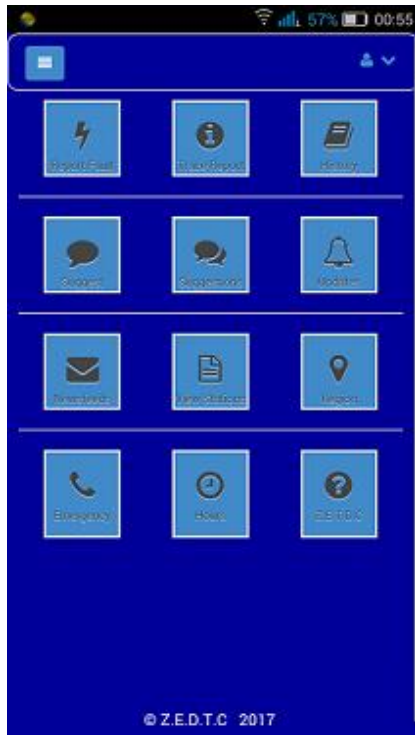
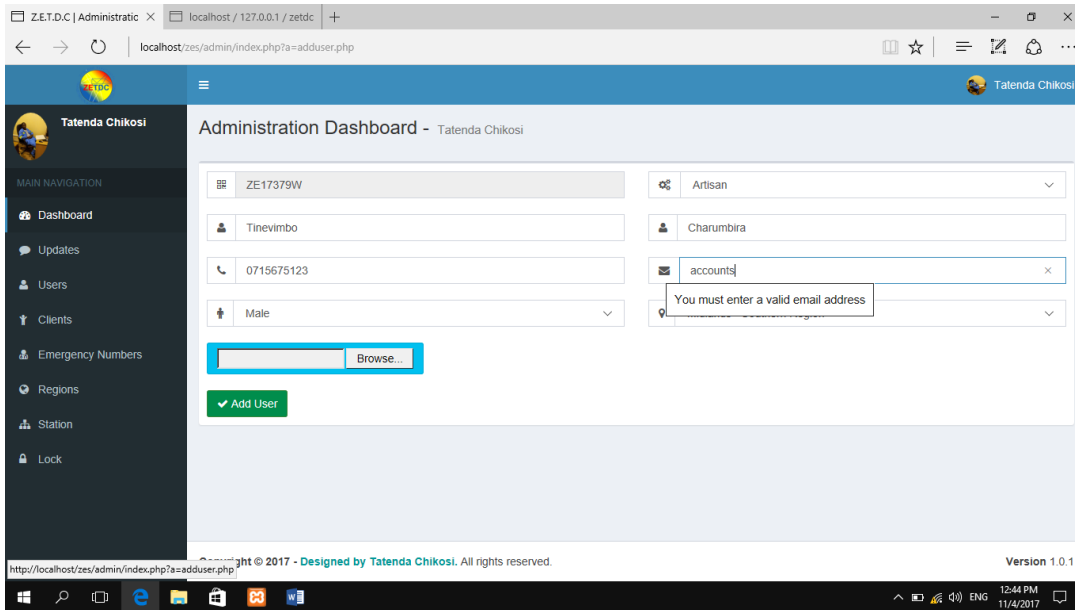


Fig 5.6 User access level

Data integrity

When entering data into the system a user should supposed to input the correct data in any given fields.



**Fig 5.7 Data integrity
Lock session**

The user may lock the system for security purposes and resumes it later on.

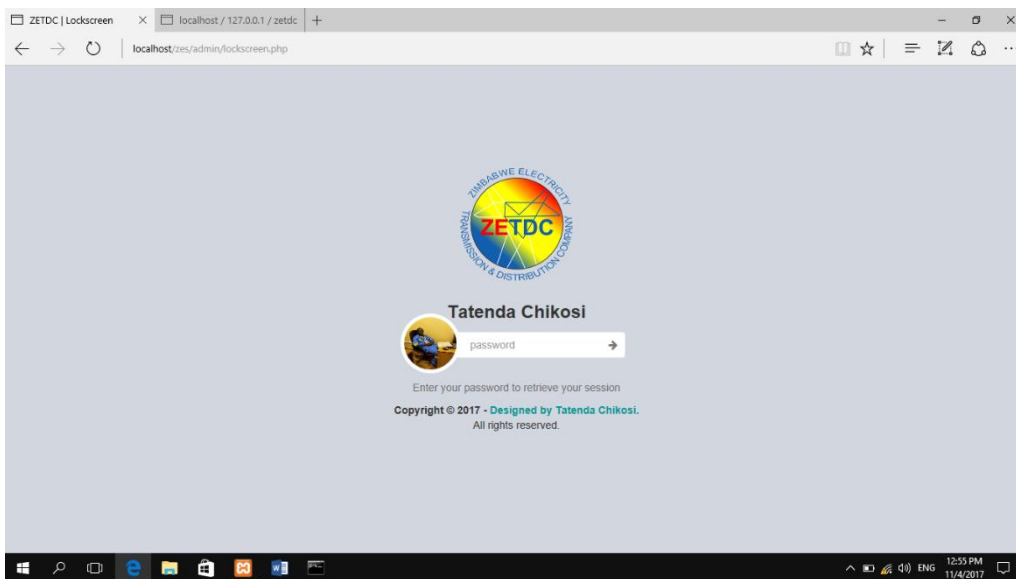


Fig 5.8 System locking

5.3.6.2 Verification

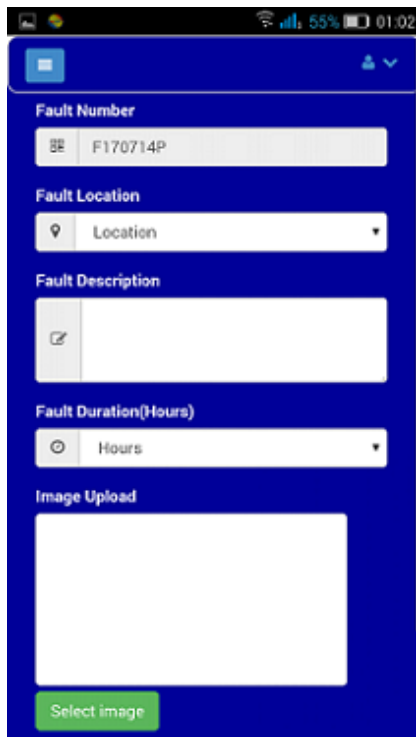
This is a process of evaluating the developed system to check if we are in the right track of creating the final product (Dennis, 2001).

It also involves the act of evaluating the software to whether its functions satisfy the conditions imposed during the beginning of the phase (Grady, 1997). The verification process is very essential because it enables the developer to check for errors before realising the

software for final use. It also checks whether the software is developed as per the specified requirement and design specification.

5.3.6.2.1 System versus Objectives Objective 1

Fault reporting – This enables the customer to report an electricity fault any time.

A screenshot of a mobile application interface for reporting a fault. The interface is displayed on a smartphone screen with a blue background. At the top, there is a status bar showing signal strength, 55% battery, and the time 01:02. Below the status bar, there is a navigation bar with a blue background and white icons for home, search, and profile. The main content area is white and contains several form fields: 'Fault Number' with a text input field containing 'F170714P'; 'Fault Location' with a dropdown menu showing 'Location'; 'Fault Description' with a text area containing a pencil icon; 'Fault Duration(Hours)' with a dropdown menu showing 'Hours'; and 'Image Upload' with a large white rectangular area and a green 'Select image' button at the bottom.

**Fig 5.9 System objective 1
Objective 2**

Change region- The system allows the customer change so as to view news and updates from that new region.



**Fig 5.10 System objective 2
Objective 3**

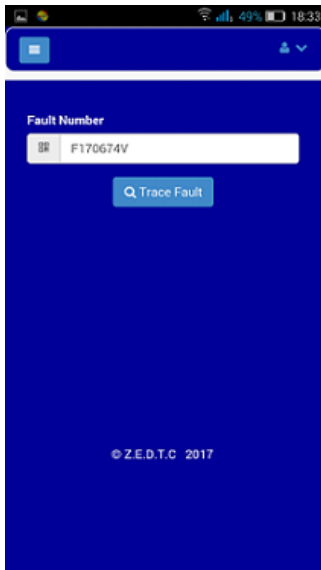
Fault progress - The system will enable the artisan to update the fault status after attending the fault.



Fig 5.11 System objective 3

Objective 4

Fault tracking – The system will provide a facility for both the depot foreman and the customer will able to track if the fault has been resolved or not and why.



**Fig 5.12 System objective 4
Objective 5**

Customer comments – The system will enable the customers to post their views, comments and suggestions as well the organisation to publish notices to its customers.

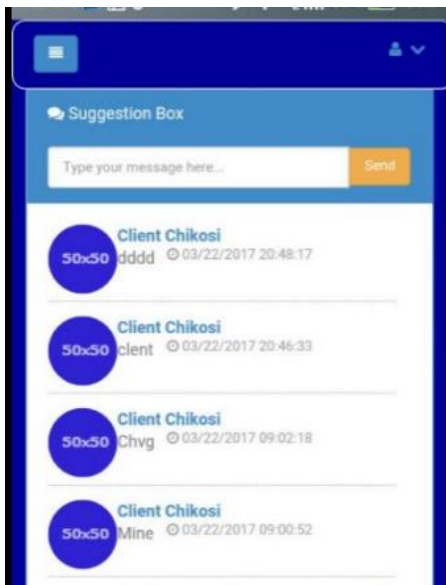


Fig 5.13 System objective 5

5.4 Installation

Shelly and Rosenblatt (2010) state that installation is process where the newly developed system is implemented in its operational environment for final use by end users. This is the

process of conversion from current to new system using various conversion strategies. The data that were used by the old system will be entered into the new system and the new system should adapt quickly. The new system will also have to be integrated with other sub packages within the organisation. There are procedures that are considered when doing installation and these are user training, data migration and system changeover.

5.4.1 User training

Training is a vital component in software development as it enables users of the system to get a better know how of how to use the system without any difficult (Mishra, 2011). This is usually done to enable users of the system to familiarise with it. Conducting a training program is essential because it reduces start-up cost, errors, and delays. All the stakeholders are trained to gain experience in using the developed system. For installation of ZETDC Fault reporting and tracking android application, a projector will be used to demonstrate how the system works to its users and the presentation will be on PowerPoint.

5.4.2 Data migration

Selah (2009) define data migration as the movement of data from the old system to be loaded in the new system. This also involves the transfer of data from one source to another (Whitten et al, 2004). The data will be loaded manually into the new system so that the processes will be done automatically. There will be risk associated with data movement from one point to another so security measures must be put in place to curb this vulnerability. Validation within the system was put in place to avoid duplication of data during data migration.

5.4.3 Changeover strategies

System changeover, as asserted by Valacich (2009), it as the process of upgrading the current system of an organisation to a better one. Upgrading this systems helps the organisation to optimize their efficiency and remain competitive. The organisation has to choose the appropriate one from the various change over strategies.

5.4.3.1 Direct changeover

Gibson and Hughes (2001) define direct changeover technique as the immediate replacement of current system to a new system. This is usually done over a weekend or after working hours. In this technique, the entire current system is replaced and as soon as the new system is in operation, the old system is shut down. This is a high risk technique because if something goes wrong, reverting back to the old system usually is impossible. As a result of

this, important data will be lost. This system is suitable where a failure in the system is not that critical for the organisation. This is the quickest of fastest changeover strategies.

5.4.3.2 Parallel changeover

Gibson and Hughes (2001) define parallel changeover as the process where the new system runs simultaneously with the old for a given period of time. This changeover is vital because it is less risk as compared to others. This means that if there is something went wrong to the new system will have to reverse back to the old system thus its original state. However, this technique is costly because they will be two system running at the same time. The process can be quite time consuming to the organization.

5.4.3.3 Phased changeover

Duffy (2011) define phased changeover strategy as the technique where the new system is implemented one stage at a time. This means the new system will be installed in phases which is useful as error will be discovered quickly. The process is quite time consuming as many phases will be installed one after the other. This technique is effective as errors and failures are limited only to the implemented modules and hence can be controllable.

5.4.3.4 Recommended changeover strategy

After having an analysis of the above changeover strategy, the ZETDC management chose to adopt the direct changeover after considering all the related risks. The data used in the old system will be entered in the new system. This was also termed to be a cheaper and fast method of conversion.

5:5 Maintenance

Valacich et al. (2012) define maintenance as the process of monitoring, evaluating and modifying the system to ensure it meets all the user requirements during its operational time and able to adapt any changes to the system.

This process is continuous as it must able to save its intended purpose over a long period of time. In addition, Somerville (2007) went on to argue that maintenance should be done periodically thus on monthly basis to examine if they are any bugs or errors encountered with the system. It can be achieved through corrective, adaptive, perfective maintenance strategies.

5.5.1 Corrective maintenance

Somerville (2007) define corrective maintenance as the process of getting rid of errors within the system upon execution. Changes to the system will take place if any errors or bugs discovered within the system during its operational time. This type of maintenance is

essential in all systems because it enables the detection and removal of bugs within the code that affects the running of the system. Errors within the system can be in form of data inconsistency, produce incorrect reports and change in data formats. As a result of this, users will have problems when using the system. Corrective maintenance is done to ensure that the system is free from errors or bugs that might affect its performance.

5.5.2 Adaptive maintenance

Shelly and Rosenblatt (2011) adaptive maintenance define as the process of checking whether the system have adapted the operational environment. The system must able to operate as intended so that it meets all the stated user requirements. This is very vital maintenance strategy because if the system fails to adapt to the operational environment then the system will fail. Changes to the system must be done if the system is not able to cope with the available environment. Adaptive maintenance is done to ensure that the system is free from errors or bugs that might affect its performance.

5.5.3 Perfective maintenance

Somerville (2004) define perfective maintenance as a process that is taken to ensure efficient within the system during its operation. This is done to incorporate new technique that may be introduced within the system. As a result of this, the organization will have efficiency and effectiveness in its operations as the system will maintained regularly. The software will be improved due the continuous maintenance that will be done frequently. Errors are slightly to occur because maintenance test will be done to prevent harm within the system. Its main aim is to analyse and identify any areas that are likely to be hindered by bugs within the system and able to rectify them. Preventive maintenance might include an antivirus should be always updated and enforce strong security policies.

5.5.4 Recommendations for maintenance

A schedule has been drawn up for the maintenance of ZETDC Fault reporting and tracking android application so that it continues to serve its purpose as intended without any problems. It is recommended that ZETDC should use corrective maintenance as its maintenance strategy and this must be done after every two (2) weeks to ensure its effectiveness as well as greater efficiency. The antivirus must be up to date every time prevent emerging viruses.

5.6 Recommendations for future development

Having finished the implementation stage, the analyst managed to notice weaknesses that needed to be considered with future developers. This is useful because in future they will be no problems as all be known by every stakeholder. The recommendations are stated below:

- In future, the management of ZETDC should consider and accepts views and ideas from low level employees not to consider top managers only. The process must be communicated to everyone within the organisation.
- It is recommended that security policies should be put into consideration and passwords should be changed on monthly basis to ensure effective security system.
- It is recommended that the ICT team should link or incorporate the system with other system within the organisation. This enable the organisation to be able to handle or manage all the system easily.
- The ICT team is recommended also in the future to make the system to be available on the web so that it can be accessed by everyone without android devices.

5.6.1 Recommendations to stakeholders

- The users are recommended to stick to the basics learnt during the training, and should they face any problems, they should consult the system administrator. The users should not disclose their password to any unauthorized personnel or let anyone temper around with the system.
- The system users should also consult the user manual to familiarize themselves with the system. The users can simply use the system with minimal training because of the friendly user interface but if they do not know how to exactly go about the system, then the user manual will be productive in information on how to fully utilize the system.

5.7 Conclusion

The new system has managed to go through all the stages from the introduction to the implementation stage without any difficult. All the system requirements as well the objectives were meet and the system is working satisfactory. After this phase, the project will be deemed a successful one. The ZETDC Fault reporting and tacking android system is an important system as it improves the quality of fault reporting thus a fault is reported and the feedback is instantly as soon as the fault is reported. However, customers will be greatly satisfied as they will be able to post the comments and provide their views to the organisation. Various test methods were done and errors were identified and rectified as soon possible and also various changeover strategies were discussed. Maintenance strategies and recommendations were also addressed in this chapter as it will be done on regular basis to improve system efficiency.

REFERENCE LIST

Buelens, M. Sinding, K. and Waldstrom, C. (2011) Organizational Behaviour. 4th edition. Berkshire: McGraw Hill

Butt, C. (1994) (Systems Analysis and Design for beginners) McGraw-Hill, Minnesota

Cwalina, K. and Abrams, F. (2005) Framework Design Guidelines: Conventions, Idions and Patterns for .NET Libraries, Addison-Wesley: London

Dubois, P. L. Jolibort, A. and Muhlbucher, H. (2007) Marketing Management. New York: Palgrave Macmillan.

Dreyer, M. (2004) Improving risk communication, National Academy of Sciences: Washington, D.C

Dennis, A. (2001) Systems Analysis and Design, 2nd edition .London

Edwards, P. (1993) Systems Analysis and Design, McGraw Hill: Singapore

Drury, C (2000) Management and Cost Accounting, 5th Edition, Tax Mann: Singapore

Gibson, M.L and Hughes, C.T (1994) Systems Analysis and Design, Massachusetts: International Thomson Publishing

Grady, O.J. (1997) System validation and verification CRC press, United States.

Hoffer, J.A. George, J.F. and Valacich, J.S. (2002) Systems Analysis and Design, 3rd edition, India: Pearson Education

Jawadekar, W. (2004) Software engineering, McGraw Hill: New Delhi

Kiyoko, C.F. and Larson, W.E. (2009).Project Management 4th Edition, McGraw Hill/Irwin, New York

Kendra, P. (2008) Management Information Systems, 6th Edition, McGraw Hill: Chicago

Laudon, K.C. and Laudon, J.P. (2013) Management Information Systems, 12th edition, United Kingdom: Pearson Education

Lucey, T. (1996) Costing 5th Edition, Ashford Colour Press Ltd, Nottingham.

Lewis, J. (2006) Fundamentals of Project Management 3rd Edition, Worksmart Publishing Company.

Lucas, H.C (1997) Information Technology for Management. 6th edition. Singapore: McGraw-Hill

McFeland, D. (2010) Dreamweaver CS5, Poqure Press Orielly: USA.

McMullan, J. (2009) Systems Analysis, (7th Edition), Prentice-Hall: London.

Marakas, G.M. and O'Brien, J.A. (2013) Introduction to Information Systems, 6th edition, New York: McGraw-Hill

Mishra, J. (2011), Software Engineering, India. : Pearson Education

Mall, R. (2009) Fundamentals of Software Engineering.3rd Ed. New Delhi. : PHI Learning Private Limited

Mikoluk, B. (2013) Research Methods for Business Students, 3rd Edition, Pearson Education, Singapore

Meloni, J. (2004) PHP5. Canada Thomson Course: Technology.

O'Brien, J.A. (2005) Introduction to Information Systems, 12th edition, New York: McGraw-Hill

Puntambekar, A.A. (2007) Software Engineering. 1st Ed.: Technical Publications Pune

Pressman, R.S. (2005) Software Engineering. 6th edition. New York: McGraw-Hill

Randall, H. (1996) Advanced Level Accounting. 3rd edition. Great Britain: Ashford Colour Press Limited

Scot, C. (1990) Database principles, Prentice Hall, United States of America

Saleh, K. (2009) Software Engineering, J Ross Publishing, Inc.

Shelly, G. and Rosenblatt, H. G., (2010). Systems analysis and design, 8th Edition Cengage Learning: Canada

Somerville, I. (2007) Software Engineering, Person: London

Whitten, J.L, Bentley, L.D and Dittman, C.K (2004) Systems Analysis and Design Methods, 6th edition, New York: McGraw-Hill

APPENDICES

APPENDIX A: USER MANUAL

The user manual is a document that is designed to help the users on how to use the system without any difficulties. It consists of steps and procedures that guides users to navigate from one system module to another throughout the system. It gives direct to the users and helps in case where the system reacts in an unusual manner.

GETTING STARTED

The ZETDC Fault reporting tracking system is an android application that enables clients to report their faults on the mobile phone anytime. The depot foreman will then assign the task to the artisan to attend the fault. The customers can as well post suggestions on the application. The customers are able to trace their fault using a fault number to if the fault has been resolved or not. The other part, the users within the organisation such as the administrator, depot foreman will access the backend and they use this site to log on www.zetdc.co.zw

The login

There are two entries or text fields labelled **username** and **password** where the user should fill in, also two buttons labelled **login** and **client account** where the user click to proceed. The login platform grant access only to those with privileges, for instance the. Access is granted through authenticating users by providing passwords and usernames.

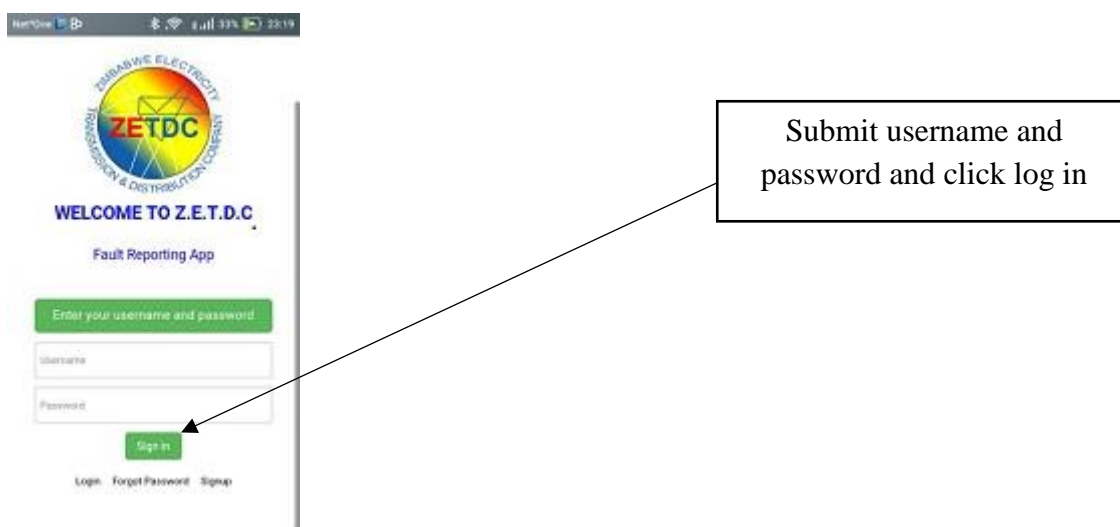


Fig A1: ZETDC Login form

Upon clicking the correct credentials, staff members and the admin are given access into the system with full access of all services given to the admin and staff limited within given parameters. If the user enters wrong login credentials, an error message will pop up as shown on.

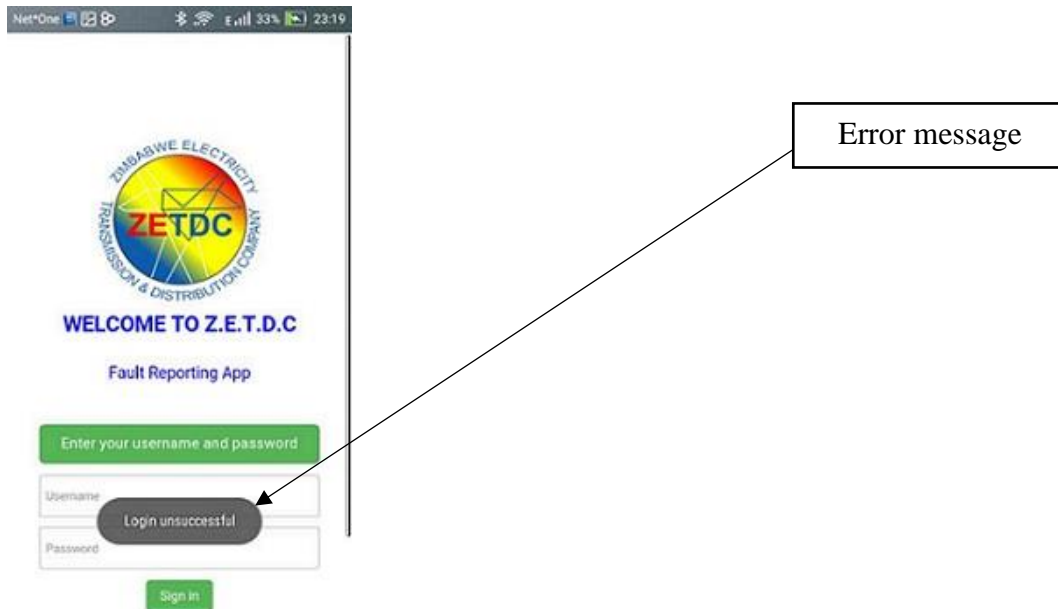


Fig A2: Login failed

Creating user account

This is where the client creates an account so as to gain access to the system.



Fig A3: User registration

After having successfully completed the registration process the client can go and put the username and the password and then logs in. If log in succeeded then all services will be displayed in the user page. This is where the user can report a fault, post suggestions and many more. The picture below shows customer page:

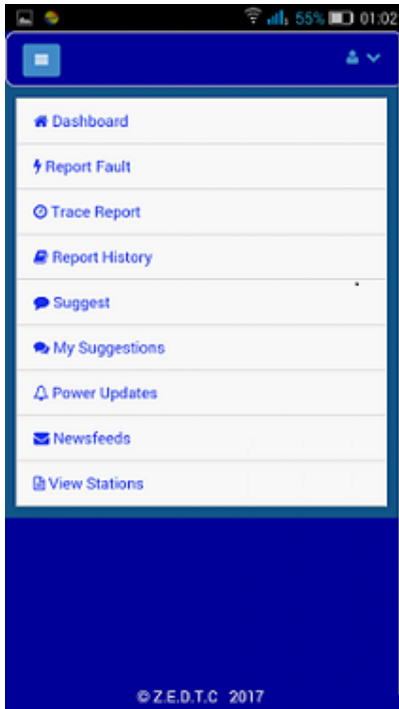


Fig A4: Customer home page

Customers can report their faults, trace faults, make suggestions and as well as to view newsfeeds and power updates. The diagram below shows the portal for reporting a fault.

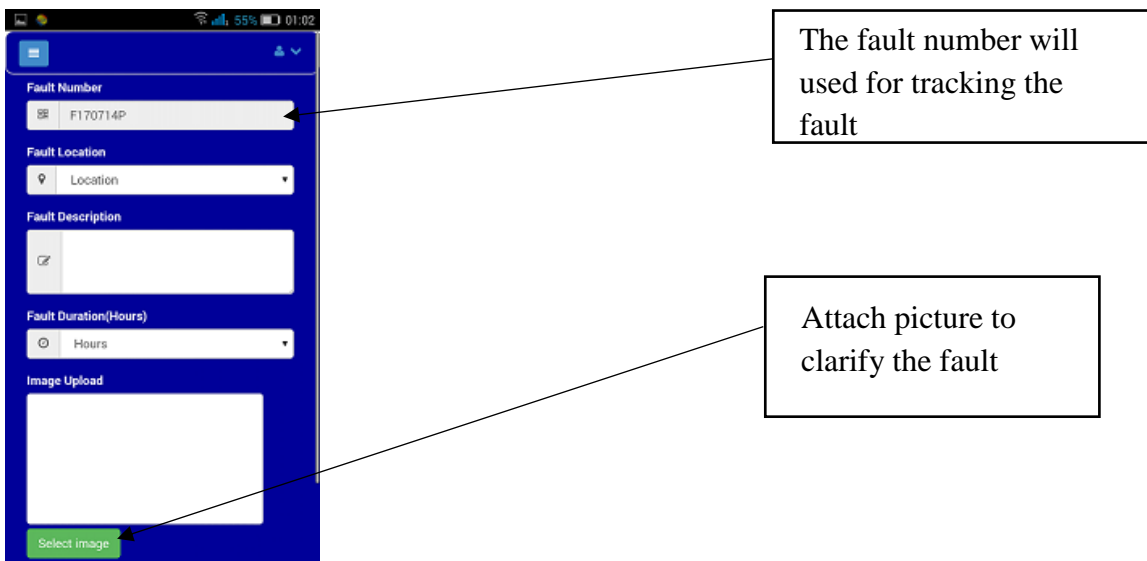


Fig A5: Fault reporting portal

The page in fig A6 shows the response from the artisan after resolving a fault. Thus updating fault status.

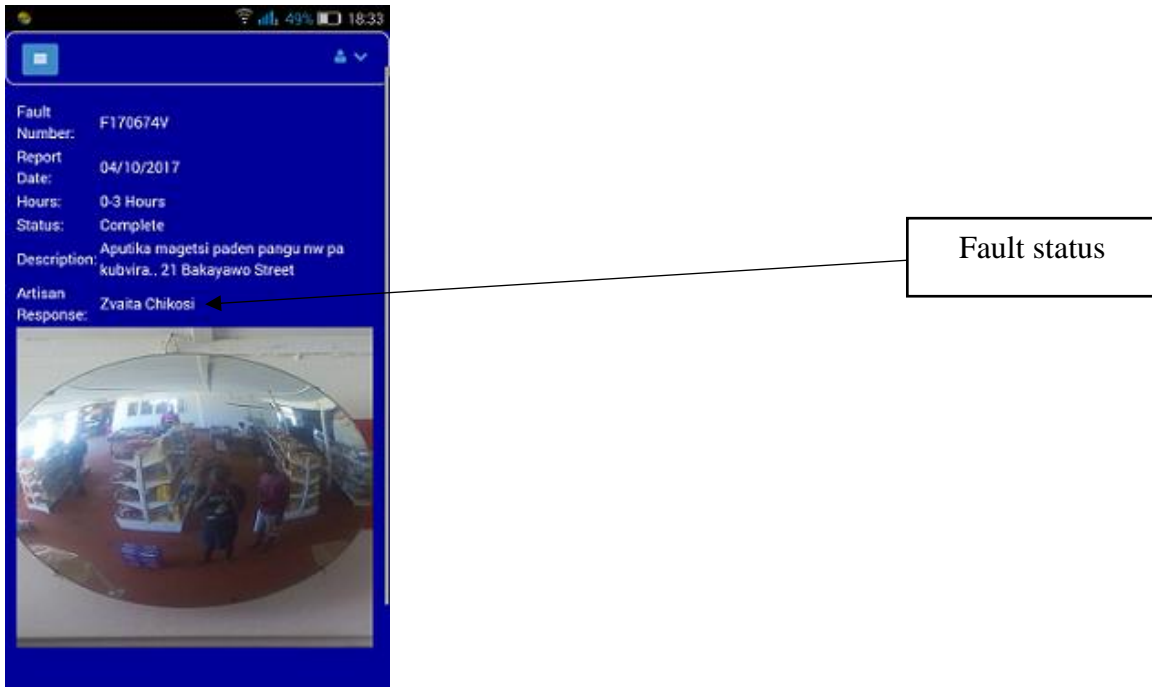


Fig A6: Fault response

The customer and the depot foreman can trace faults on this page. They will both check the progress of the fault.



Fig A7: Tracing a fault

The system allows the customer to change Region for example southern region to northern region. This helps if the customer re-locates to another city thus the customer can now able to view updates of the province.

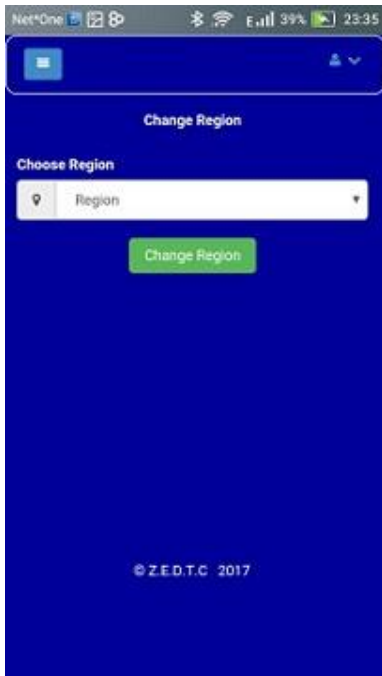


Fig A8: Change region

The page below shows the page where the customer will be posting suggestions. These suggestions will be viewed by the administrator for decision making purposes.

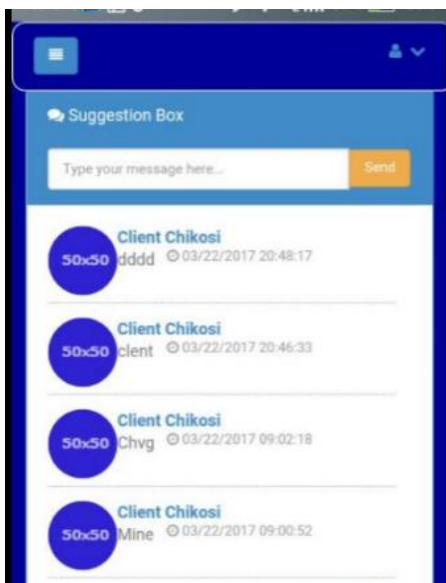


Fig A9: Suggestion chat

The customer can view stations that are available at the organisation in that region.

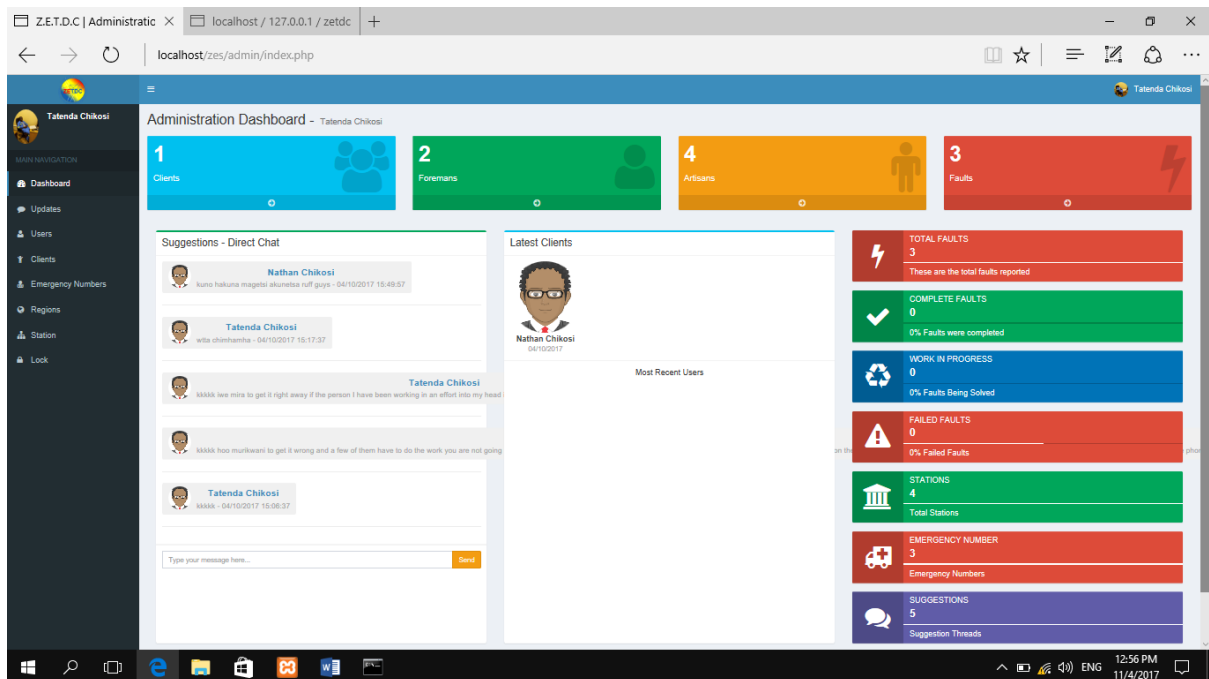


Station Name	Region	Address
Kwekwe Depot	Midlands	771, Bulawayo
Redcliff Depot	Midlands	100, Bulawayo
Msasa Station	Midlands	100, Bulawayo
Mbine Station	Midlands	100, Bulawayo

© Z.E.D.T.C 2017

Fig A10: Stations

The home page below shows the information that will be accessed by the administrator. The admin monitors all the information flowing within the system. The admin is responsible for adding users, post update, add stations as well as adding emergency numbers.



Administration Dashboard - Tatenda Chikosi

- 1 Clients
- 2 Foremans
- 4 Artisans
- 3 Faults

Suggestions - Direct Chat

Latest Clients

Most Recent Users

TOTAL FAULTS: 3
These are the total faults reported

COMPLETE FAULTS: 0
0% Faults were completed

WORK IN PROGRESS: 0
0% Faults Being Solved

FAILED FAULTS: 0
0% Failed Faults

STATIONS: 4
Total Stations

EMERGENCY NUMBER: 3
Emergency Numbers

SUGGESTIONS: 5
Suggestion Threads

Fig A11: Admin home page

The page below shows the admin giving the role of an artisan to a particular user.

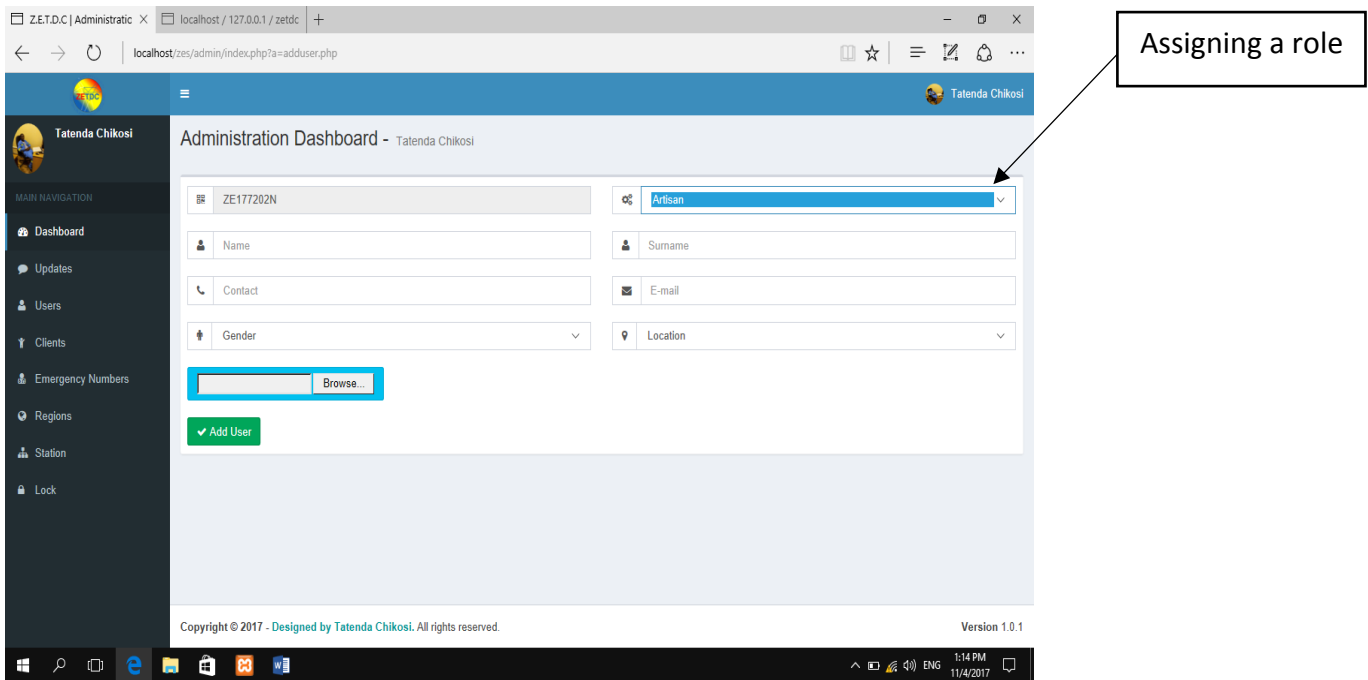


Fig A12: Assign a role

The page below shows the login successful if the admin or the deport foreman puts correct credential to logon:

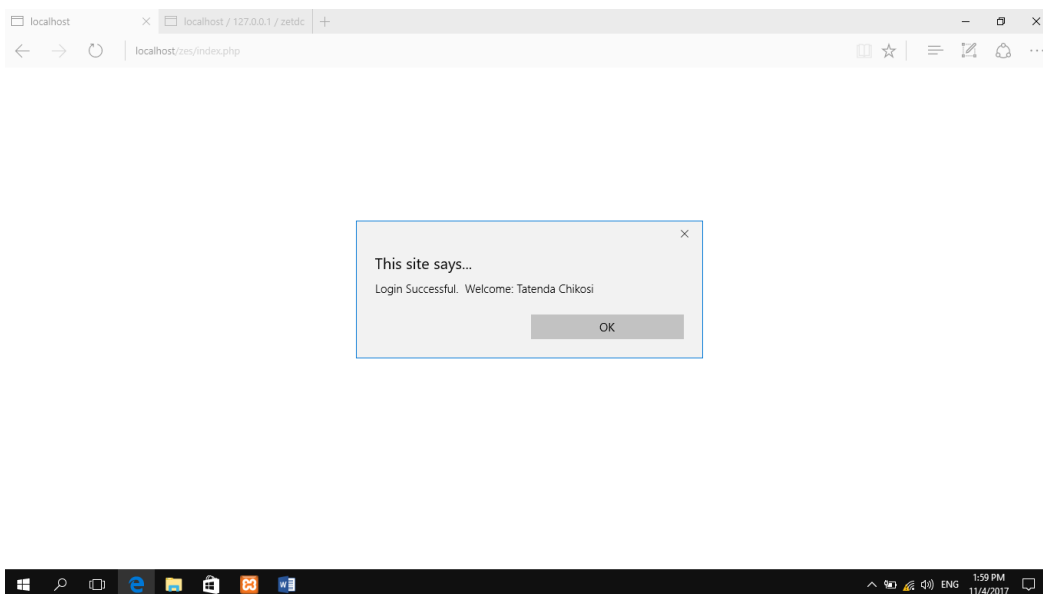


Fig A13: Admin login successful

The page below shows the admin deleting a certain user. The admin has got the right to do so.

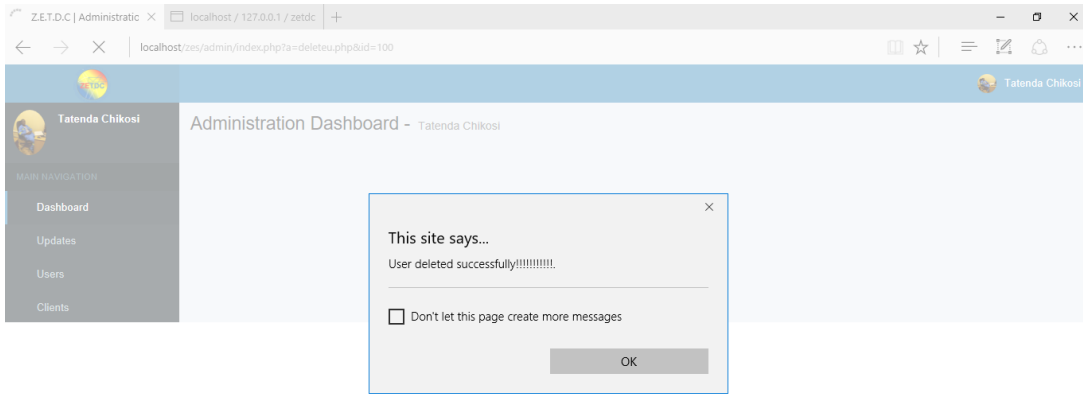


Fig A14: Deleting a user

The page below show the admin creating a user and as well assigning a role to that user.

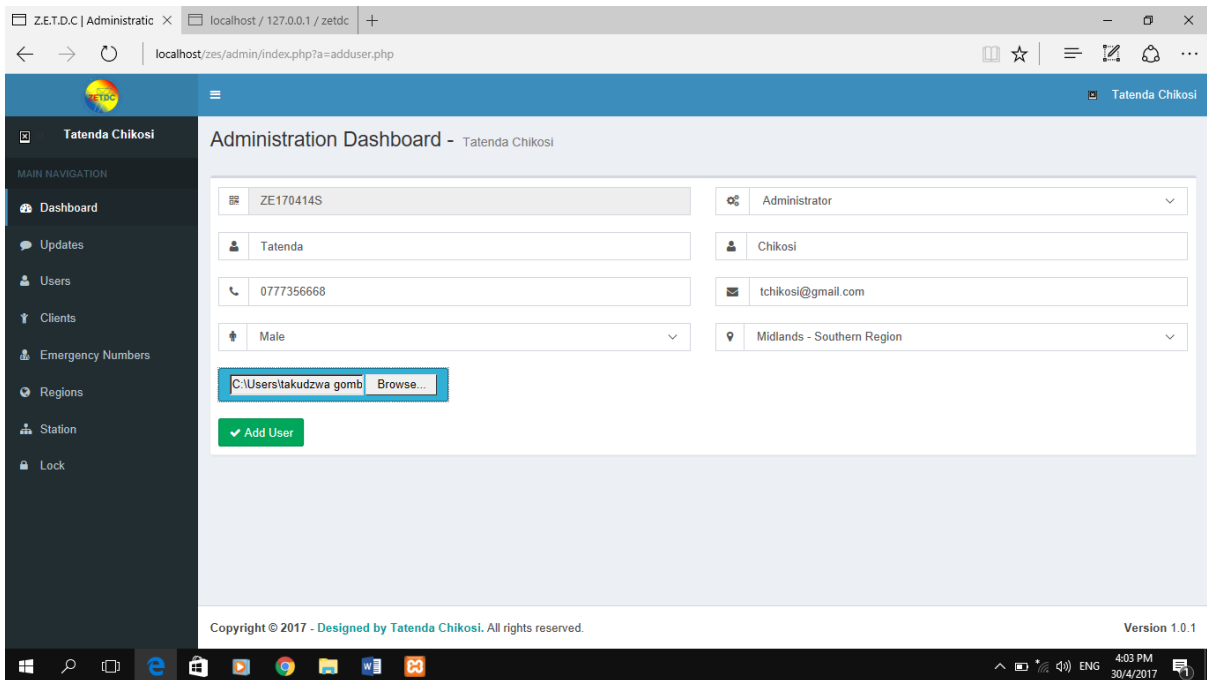


Fig A15: Adding a user

The following diagram shows an error message that comes out if the admin wants to add a user without filling in some fields.

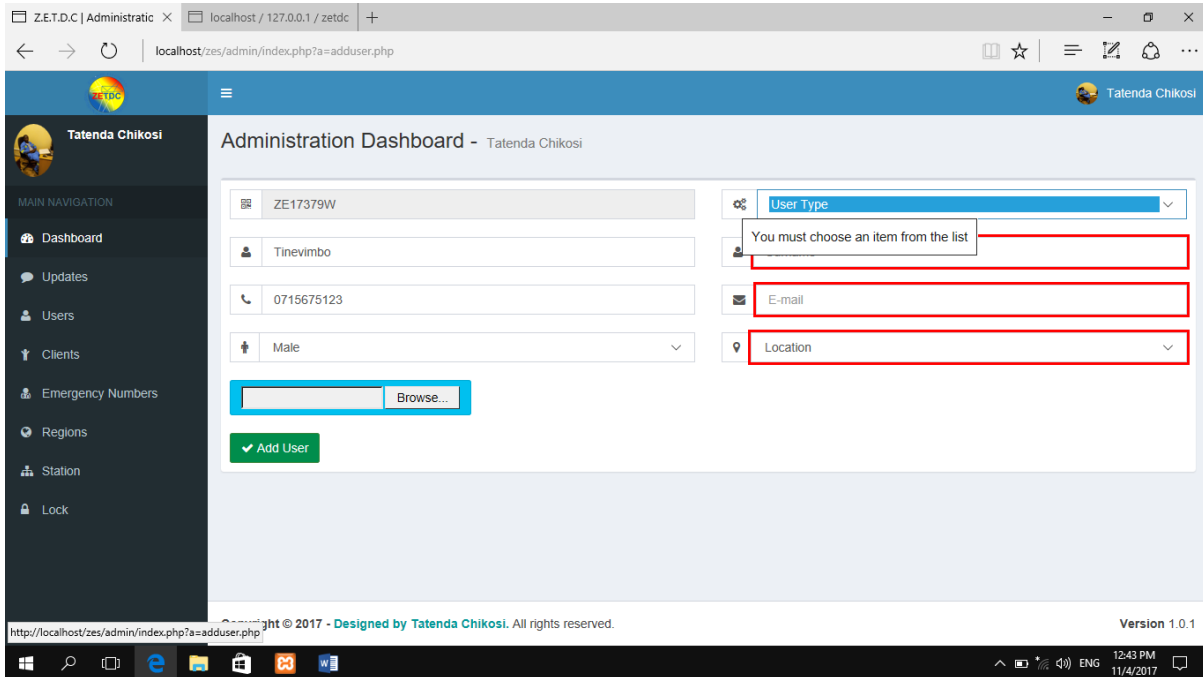


Fig A16: Error of fill in

The user may lock the system for security purposes and resumes it later on.

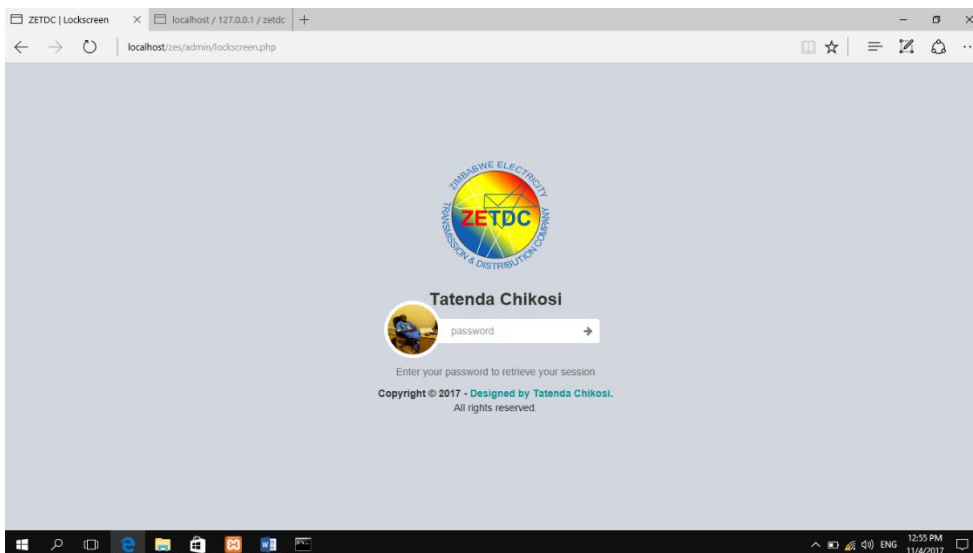


Fig A17: Lock session

The page below shows the faults reported by customers that are viewed by the depot foreman.

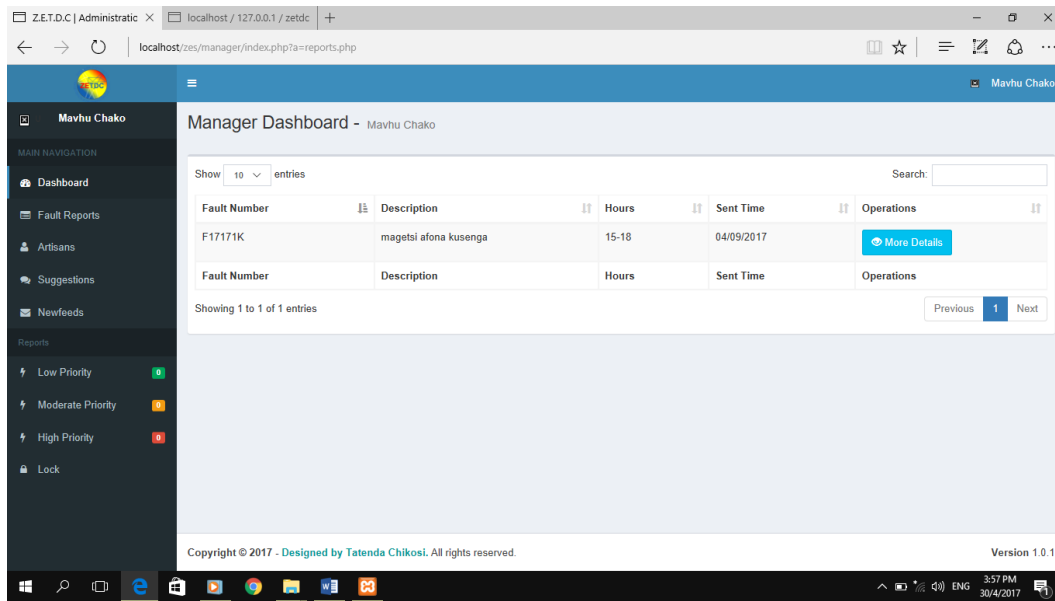


Fig A18: Fault reports

The page below shows the list of users that are in the database and these are added by the admin. The users have got different access level

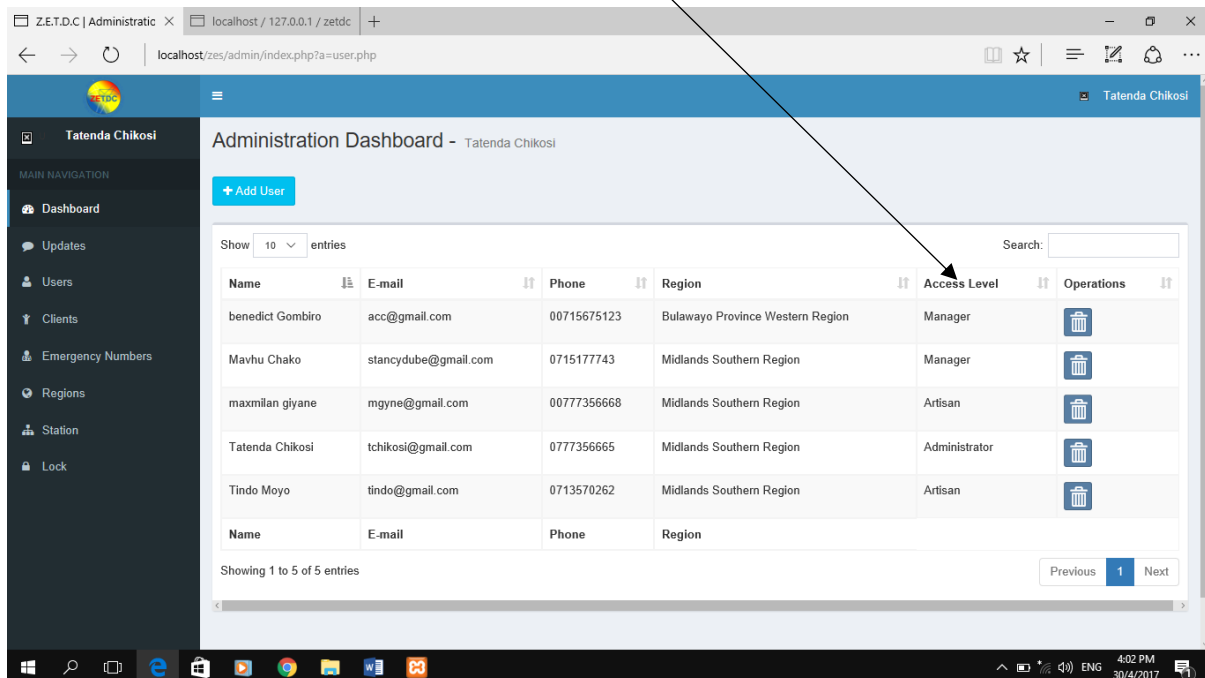


Fig A19: Users

APPENDIX B: INTERVIEW CHECKLIST

PLACE.....

DATE.....

DEPARTMENT INTERVIEWED

STORES DEPARTMENT

1. How do you view the current system in your own opinion

.....
.....
.....
.....

2. What can you say about the current system's performance?

.....
.....
.....

3. How do you restrict access to files which contain sensitive information?

.....
.....
.....

4. What actions can't you do with the current system?

.....
.....

5. What would you want the new system to incorporate?

.....
.....

6. Any recommendations or suggestions to the development of the new system?

.....
.....
.....
.....
.....

7. Explain the major problem that you have faced so far with the old system?

.....
.....
.....
.....
.....

APPENDIX C: QUESTIONNAIRE CHECK LIST

PLACE

DATE.....

EMPLOYEES

Please tick where it is applicable

1. Do you think the current system is reliable, efficient and contributing to your organizational objectives and goals effectively?

Excellent system

Fair

Ineffective

2. Do you often experience very busy days and big workloads during your operations?

Very often

infrequently

All the time

3. Have some employees complained about the current system's performance?

Yes

No

A few

4. Have there been complaints from customers about your services?

Yes

No

A few

12. Are you pleased with the processing speed of the current system?

Yes

No

APPENDIX D: OBSERVATION SCORE SHEET

OBSERVATION SCORE SHEET

NAME OF THE OBSERVER

DATE OF OBSERVATION

TIME OF OBSERVATION

PLACE OF OBSERVATION

.....

OBJECT BEING OBSERVED

.....

OBSERVATION.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

CONCLUSION

.....

.....

.....

.....

.....

.....

APPENDIX E: SNIPPET OF CODE

LOGIN

MainActivity.java

```
package com.javacodegeeks.android.loginapp;

import android.graphics.Color;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity {

    private EditText username;
    private EditText password;
    private Button login;
    private TextView loginLockedTV;
    private TextView attemptsLeftTV;
    private TextView numberOfRemainingLoginAttemptsTV;
    int numberOfRemainingLoginAttempts = 3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setupVariables();
    }

    public void authenticateLogin(View view) {
        if (email.getText().toString().equals("admin") and and
password.getText().toString().equals("admin")) {
            Toast.makeText(getApplicationContext(), "Hello
admin!",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Seems like
you 're not admin!",
                Toast.LENGTH_SHORT).show();
            numberOfRemainingLoginAttempts--;
            attemptsLeftTV.setVisibility(View.VISIBLE);

            numberOfRemainingLoginAttemptsTV.setVisibility(View.VISIBLE);

            numberOfRemainingLoginAttemptsTV.setText(Integer.toString(numberOfR
emainingLoginAttempts));

            if (numberOfRemainingLoginAttempts == 0) {
                login.setEnabled(false);
                loginLockedTV.setVisibility(View.VISIBLE);
                loginLockedTV.setBackgroundColor(Color.RED);
                loginLockedTV.setText("LOGIN LOCKED!!!");
            }
        }
    }
}
```

```

    }

    private void setupVariables() {
        email = (EditText) findViewById(R.id.usernameET);
        password = (EditText) findViewById(R.id.passwordET);
        login = (Button) findViewById(R.id.loginBtn);
        loginLockedTV = (TextView)
findViewById(R.id.loginLockedTV);
        attemptsLeftTV = (TextView)
findViewById(R.id.attemptsLeftTV);
        numberOfRemainingLoginAttemptsTV = (TextView)
findViewById(R.id.numberofRemainingLoginAttemptsTV);

        numberOfRemainingLoginAttemptsTV.setText(Integer.toString(numberOfR
emainingLoginAttempts));
    }

```

ADD USER

```

<?php

if(isset($_POST['create']))
{
    if(strlen($_POST['name'])<3)
    {
        echo("<script language=javascript>window.alert('Your name should be more
than three characters'); window.location= 'index.php?a=adduser.php';</script> ");
        exit;
    }
    if(strlen($_POST['surname'])<3)
    {
        echo("<script language=javascript>window.alert('Your surname should be
more than three characters'); window.location= 'index.php?a=adduser.php';</script> ");
        exit;
    }

    $query = "SELECT * from users where empnum='".$_POST[reg]' ";
    $result = mysqli_query($dbc, $query);
    $row = mysqli_num_rows($result);
    if($row == 1)
    {
        ?><script language="javascript"> alert("Employee Number already in use"); location
= 'index.php?a=user.php'</script> <?php
        exit;
    }

    $query = "SELECT * from users where email='".$_POST[email]' ";
    $result = mysqli_query($dbc, $query);
    $row = mysqli_num_rows($result);
    if($row >0 )
    {
        ?><script language="javascript"> alert("E-mail already in use"); location =
'index.php?a=user.php'</script> <?php
        exit;
    }

```

```

}
    $nquery = "SELECT * from users where contact='$_POST[contact]' ";
    $nresult = mysqli_query($dbc, $nquery);
    $nrow = mysqli_num_rows($nresult);
    if($nrow >0)
    {
        ?><script language="javascript"> alert("Contact already in use"); location =
'index.php?a=user.php'</script> <?php
        exit;
    }
    if (!isset($_FILES['image']['tmp_name']))
    {
        echo("<script language=javascript>window.alert('Please Check Image Upload....It
Failed to attach file. '); window.location= 'index.php?a=adduser.php';</script> ");
        exit;
    }
    else
    {
        $file=$_FILES['image']['tmp_name'];
        $image= addslashes(file_get_contents($_FILES['image']['tmp_name']));
        $image_name= addslashes($_FILES['image']['name']);
        $image_size= getimagesize($_FILES['image']['tmp_name']);

        if ($image_size===FALSE)
        {
            echo("<script language=javascript>window.alert('Oops This is not an
image. '); window.location= 'indexad.php?a=screates.php';</script> ");
            exit;
        }
        else
        {
            $nguva=time();
            move_uploaded_file($_FILES["image"]["tmp_name"],"../uploads/" .
$_FILES["image"]["name"]);
            $location="uploads/" . $_FILES["image"]["name"];
            $result = mysqli_query($dbc, "INSERT INTO `users`(`name`,
`surname`, `script`, `contact`, `email`, `sex`, `access`, `username`, `password`, `state`,
`region`, `datecreated`, `addedby`, `empnum`)values

            ($_POST[name]',$_POST[surname]','$location',$_POST[contact]',$_POST[email]','
$_POST[gender]','$_POST[access]','$_POST[reg]',newstaff,'0',$_POST[region]','$nguva','$i
dd',$_POST[reg]') ")or die (mysqli_error($dbc));

            if(!$result)
            {
                echo("<script
language=javascript>window.alert('Oops failed to create user please try again. ');
window.location= 'index.php?a=adduser.php';</script> ");
                exit;
            }

```

```

        else
        {
            echo("<script language=javascript>window.alert('User created
successfully!!!!!!!!!!!!!!'); window.location= 'index.php?a=user.php';</script> ");
            exit;
        }
    }
}
?>

```

```

<!-- Main content -->
<section class="content">

```

```

<!-- row -->

```

```

<div class="row">

```

```

    <div class="box">

```

```

    <div class="box-body ">

```

```

        <form action="" method="post" class="horizontal-form"
enctype="multipart/form-data" name="addroom" >

```

```

            <div class="row">

```

```

                <div class="col-md-6">

```

```

                    <div class="input-group">

```

```

                        <span class="input-group-addon"><i class="fa fa-qrcode"></i></span>

```

```

                        <input class="form-control" name="reg" readonly value="<?php

```

```

error_reporting(0);

```

```

$characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

```

```

$numbers = "0123456789";

```

```

$length = 4;

```

```

        for ($p = 0; $p < $length; $p++) {

```

```

            $code = $characters[mt_rand(0, strlen($characters))];

```

```

            $codee .= $numbers[mt_rand(0, strlen($numbers))];

```

```

            $date=date('y');

```

```

            $reg= "ZE".$date.$codee.$code;

```

```

        }

```

```

        echo $reg;

```

```

?>"type="text" required>

```

```

    </div>

```

```

</div>

```

```

<!-- /.col -->

```

```

<div class="col-md-6">

```

```

    <div class="input-group">

```

```

        <span class="input-group-addon"><i class="fa fa-gears"></i></span>

```

```

        <select class="form-control" name="access" required>

```

```

            <option value="">User Type</option>

```

```

            <option value="0">Administrator</option>

```

```

        <option value="1">Artisan</option>
        <option value="3">Manager</option>
    </select>
</div>
</div>
<!-- /.col -->
</div><br>
        <div class="row">
<div class="col-md-6">
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-user"></i></span>
        <input class="form-control" placeholder="Name" name="name" type="text"
required>
    </div>
</div>
<!-- /.col -->
<div class="col-md-6">
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-user"></i></span>
        <input class="form-control" name="surname" placeholder="Surname" type="text"
required >
    </div>
</div>
<!-- /.col -->
</div><br>
        <div class="row">
<div class="col-md-6">
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-phone"></i></span>
        <input class="form-control" placeholder="Contact" name="contact"
pattern="\d{10}" required type="tel">
    </div>
</div>
<!-- /.col -->
<div class="col-md-6">
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-envelope"></i></span>
        <input class="form-control" placeholder="E-mail" name="email" type="email"
required>
    </div>
</div>
<!-- /.col -->
</div><br>
        <div class="row">
<div class="col-md-6">
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-male"></i></span>
        <select class="form-control" name="gender" required>
            <option value="">Gender</option>
            <option >Male</option>

```



```

        <option >Female</option>
    </select>
</div>
</div>
<!-- /.col -->
<div class="col-md-6">
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-map-marker"></i></span>
        <select class="form-control" required name="region">
            <option value="">Location</option>
            <?php
                $rs=mysqli_query($dbc,"SELECT * FROM
`regions` ORDER BY `regions`.`province` ASC ") or die(mysql_error());
                while($row=mysqli_fetch_array($rs))
                    {
                        ?>
                        <option value="<?php echo $row["id"]; ?>"><?php echo $row["province"]." -
".$row["regionname"]; ?></option>
                        <?php
                            }
                        ?>
                    </select>
                </div>
            </div>
            <!-- /.col -->
        </div><br>
        <div class="row">
            <div class="col-md-12">
                <input id="exampleInputFile" class="btn btn-info " name="image" type="file">
            </div>
            <!-- /.col -->
        </div><br>
        <button type="submit" class="btn btn-success" name="create"><i class="fa fa-
check"></i> Add User</button>
    </form>
</div>
</div>
</div>
<!-- /.row -->
</section>
<!-- /.content -->

```

ASSIGNMENT HISTORY

```

<?php
    $assignee=$_SESSION['username'];

    $nguva=date('m/d/Y')." - ".time('H:m:s');

```

```

$result = mysqli_query($dbc, "SELECT * from faults where
id='$_REQUEST[id]'" or die (mysqli_error($dbc));
$rows=mysqli_fetch_array($result);
    $fault_id=$rows['fault_id'];
    $sent_time=$rows['sent_time'];
    $fault_description=$rows['fault_description'];
    $status=$rows['status'];
if(isset($_POST['submit'])){
    $artisan_id = $_POST['artisan_id'];
    $priority = $_POST['priority'];
    $result = mysqli_query($dbc, "INSERT INTO `assignments`(`fault_id`, `artisan_id`,
`assignment_timestamp`, `assigned_by`)VALUES
('$fault_id','$artisan_id','$nguva','$assignee')" or die (mysqli_error($dbc));

    if(!$result)
        {
            echo("<script
language=javascript>window.alert('Oops failed to assign artisan please try again.');"
window.location= 'index.php?a=reports.php';</script> ");
            exit;
        }
        else
        {
            $res = mysqli_query($dbc, "UPDATE `zetdc`.`users` SET `state` =
'Assigned' WHERE `users`.`empnum` = '$artisan_id'" or die (mysqli_error($dbc));
            $res = mysqli_query($dbc, "UPDATE `zetdc`.`faults` SET
`artisan_id` = '$artisan_id' WHERE `fault_id` = '$fault_id'" or die (mysqli_error($dbc));
            echo("<script language=javascript>window.alert('Assigned
Successfully');" window.location= 'index.php?a=reports.php';</script> ");
            exit;
        }
    }
?>
<!-- Main content -->
<section class="content">
<!-- row -->
<div class="row">

    <div class="box">

        <div class="box-body ">
            <div class="box-body">
                <form action="" method="post" class="form-signin">
                    <center></center>
                    <dl class="dl-horizontal">
                        <dt>Fault Number:</dt>
                        <dd><?php echo $fault_id; ?></dd><br>
                        <dt>Priority:</dt>

```

```

<dd><select class="form-control" name="priority" required>
  <option value="">Select Priority </option>
  <option>Low Priority</option>
  <option>Moderate Priority</option>
  <option>High Priority</option>
</select></dd><br>
<dt>Artisan:</dt>
<dd><select class="form-control" required name="artisan_id">

<option value="">Currently Available </option>
      <?php
          $rs=mysqli_query($dbc,"SELECT * FROM
`users` where `access`='1' and `region`='$sender_id' and `state`='0'") or die(mysql_error());
          while($row=mysqli_fetch_array($rs))
              {
                  ?>
          <option value="<?php echo $row["empnum"]; ?>"><?php echo $row["empnum"];
?></option>
              <?php
                  }
              ?>
          </select></dd>

</dl>
<center><button class="btn text-muted text-center btn-success" name="submit"
type="submit">Assign Task</button></center>
      </form>
</div>
</div>
</div>
</div>
<!-- /.row -->
</section>
<!-- /.content -->

```